

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b>					
<b>1. REPORT DATE (DD-MM-YYYY)</b> 05/09/2007		<b>2. REPORT TYPE</b> Final Report		<b>3. DATES COVERED (From - To)</b> 05/10/2006 – 05/09/2007	
<b>4. TITLE AND SUBTITLE</b>  Center for Coastline Security Technology, Year-2				<b>5a. CONTRACT NUMBER</b> N00014-05-C-0031	
				<b>5b. GRANT NUMBER</b> N/A	
				<b>5c. PROGRAM ELEMENT NUMBER</b> N/A	
<b>6. AUTHOR(S)</b>  Stewart Glegg, William Glenn, Borko Furht, P.P. Beaujean, G. Frisk, S, Schock, K. VonEllenrieder, P. Ananthakrishnan, E. An, R. Granata, R. Coulson.				<b>5d. PROJECT NUMBER</b> N/A	
				<b>5e. TASK NUMBER</b> CLIN 0005	
				<b>5f. WORK UNIT NUMBER</b> N/A	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  Florida Atlantic University 777 Glades Road Boca Raton, FL 33431				<b>8. PERFORMING ORGANIZATION REPORT</b>	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>  Office Of Naval Research 875 North Randolph Street, Suite 1425 Arlington VA 22203				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b> O.N.R.	
				<b>11. SPONSOR/MONITOR'S NUMBER(S)</b>	
<b>12. DISTRIBUTION / AVAILABILITY STATEMENT</b> A – Approved for public release					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b> See Attached.					
<b>15. SUBJECT TERMS:</b> Underwater vehicle; Acoustic communications; Environmental assessments; Imaging sonar; Underwater navigation; Chemical sensors; High resolution cameras; Three dimensional imaging.					
<b>16. SECURITY CLASSIFICATION OF:</b> Unclassified/Unlimited			<b>17. LIMITATION OF ABSTRACT</b>  N/A	<b>18. NUMBER OF PAGES</b>	<b>19a. NAME OF RESPONSIBLE PERSON</b> Sylvie Butel
<b>a. REPORT</b> N/A	<b>b. ABSTRACT</b> N/A	<b>c. THIS PAGE</b> N/A			<b>19b. TELEPHONE NUMBER</b> (561) 297-2366



## **Report Documentation SF298**

### **Section 14 – Abstract**

The Center for Coastline Security Technology (CCST) focuses on research, simulation, and evaluation of coastal defense and marine domain awareness equipment, sensors, and components. It builds upon the existing efforts and expertise in coastal systems and sensor research at the Institute for Ocean and Systems Engineering (IOSE), the Imaging Technology Center, the Department of Computer Science and Engineering, and the University Consortium for Intermodal Transportation Safety and Security at Florida Atlantic University.

This report describes a number of projects that were carried out during year two of this program. The following projects are described in the report

- 1) Development of a Remotely Piloted, Unmanned, Untethered, Underwater Vehicle (RPUUV)
- 2) Development of Acoustic Piloting, Communications and Positioning systems
- 3) Environmental Assessment and Modeling: Monitoring Turbidity in Ports
- 4) Development of a High Resolution Imaging Sonar for Underwater Inspections
- 5) Experimental determination of the hydrodynamic/dynamic characteristics of a small underwater vehicle for port security
- 6) Hydrodynamic and Dynamic Investigations for the Development of a Small Underwater Vehicle for Underwater Hull Inspection and Harbor Survey
- 7) RPUUV Navigation and Control
- 8) Development of a Chemical Sensor system for small underwater vehicles
- 9) Development of HDMAX High-Resolution QUAD-HD Progressive Scan Electronic Camera Systems
- 10) 3D Imaging and 3D Video Technologies for Coastline Security Applications



## **Center for Coastline Security Technology**

### **Year 2: Final Technical Report**

Contract/PR No. N00014-05-C-0031

Prepared for  
US Office of Naval Research

For the Period  
10 May 2006 to 09 May 2007

Submitted by  
Stewart Glegg, William Glenn, Borko Furht, P. Beaujean, G. Frisk, S. Schock, K.  
vonEllenrider, P. Ananthakrishnan, Edgar An, R. Granata, R. Coulson

College of Engineering  
Florida Atlantic University,  
777 Glades Road  
Boca Raton, FL 33431  
561 297 3000

Submitted May 9<sup>th</sup>, 2007



## ABSTRACT

The Center for Coastline Security Technology (CCST) focuses on research, simulation, and evaluation of coastal defense and marine domain awareness equipment, sensors, and components. It builds upon the existing efforts and expertise in coastal systems and sensor research at the Institute for Ocean and Systems Engineering (IOSE), the Imaging Technology Center, the Department of Computer Science and Engineering, and the University Consortium for Intermodal Transportation Safety and Security at Florida Atlantic University.

New technologies are needed to enhance surveillance and inspections of marine activities in the coastal zone that includes major ports, small inlets, beaches, remote coastal areas, and their approaches. The task is to effectively integrate sensors with underwater, surface, and airborne autonomous and remotely operated platforms and to incorporate video and image analysis and data mining methods to quickly and effectively identify threat events.

The technologies that will be developed in this program are:

- 1) *Underwater vehicles for survey and inspection:* In the CCST program a low cost, one-man-operated, remotely-piloted and unmanned, untethered, underwater vehicle, is being developed, that will provide real-time underwater video and sonar images to a topside console. The specific applications to be addressed are underwater inspections by rapid response teams, and routine inspection activities, currently carried out by scuba divers. This technology is intended to reduce the need for divers on a 24/7 basis. During year one of the program a vehicle was developed with a tow float and an RF antenna to provide the underwater video and sonar data to a topside console. In year two a tetherless capability has been added by replacing the tow float with a high-speed acoustic modem. In addition, a high-resolution sonar system has been developed, that will be mounted on the vehicle in the third year of the program. The high-resolution sonar system will operate in side scan mode and will rotate about its axis to provide images from different aspects. The design of the sonar system is an important first step towards the overall objective of developing a high-resolution underwater images of ship hulls and port seawalls.
- 2) *High Definition Video Systems:* High-definition video cameras provide an order of magnitude improvement in field of view and/or range over those achievable with conventional video systems. They are thus a necessity for harbor surveillance; however, their implementation in this environment is limited by size and cost. At Florida Atlantic University's Imaging Technology Center, a compact super-high-definition camera (with four times the resolution of conventional high-definition video cameras) has been developed and is ready for the commercial market, the primary customers being the cinematic film industry. For the port security application there are several research issues being addressed under this program, specifically, recording the output of the camera, managing the high data output rate of the camera, testing the camera in the marine environment, and combining a

pair of the cameras with a matched pair of digital video projectors for real-time 3D surveillance. The test and evaluation issue will be addressed by the ITC in collaboration with NAVSEA Carderock's South Florida Test Facility, which has towers overlooking Port Everglades, and the adjacent inlet, already used by the USCG for video surveillance. Software enhancement of 3D imaging using the HDMAX camera will be addressed by Florida Atlantic University's Department of Computer Science and Engineering.

In this report the details for year two of this program will be presented. The following projects are described

- The Remotely Piloted, Unmanned, Untethered, Underwater Vehicle (RPUUV)  
PIs Dr. S. Glegg
- Acoustic Piloting, Communications and Positioning  
PI: Dr. P. Beaujean
- Environmental Assessment and Modeling: Monitoring Turbidity in Ports  
PI: Dr. George V. Frisk
- Development of a High Resolution Imaging Sonar for Underwater Inspections  
PI: Dr. Steven Schock
- Experimental determination of the hydrodynamic/dynamic characteristics of a small underwater vehicle for port security  
PI: Dr. von Ellenrieder
- Hydrodynamic and Dynamic Investigations for the Development of a Small Underwater Vehicle for Underwater Hull Inspection and Harbor Survey  
PI: P. Ananthakrishnan
- RP UUV Navigation and Control  
PI: Dr. Edgar An
- Chemical Sensors  
PI: Dr. Richard Granata
- HDMAX High-Resolution QUAD-HD Progressive Scan Electronic Camera Systems  
PI: Dr. W. Glenn,
- 3D Imaging and 3D Video Technologies for Coastline Security Applications  
PI: Dr. B. Furht

## TABLE OF CONTENTS

Abstract	
List of Figures	
List of Tables	
Executive Summary	
1.0 INTRODUCTION	
1.1 Overview	
1.1.1 Background	
1.1.2 Technical Objectives	
2.0 THE REMOTELY PILOTED, UNMANNED, UNTETHERED, UNDERWATER VEHICLE (RPUUV)	
2.1 Background	
2.2 Development of a Remotely Piloted Unmanned Underwater Vehicle	
PI: Dr. Stewart Glegg	
2.2.1 Summary	
2.2.2 Introduction	
2.2.3 Re-design of the RPUUV to Support an Acoustic Modem, High Resolution Imaging Sonar & Explosives Detection Payloads (Tasks 3.1, 3.2, & 3.3)	
2.2.3.1 Acoustic Modem Integration	
2.2.3.2 Payload Integration Design	
2.2.4 Additional New Sensor Systems & Modifications (Tasks 3.4 & 3.5)	
2.2.4.1 LED Lighting	
2.2.4.2 Pressure Sensor	
2.2.4.3 Altimeter	
2.2.4.4 Obstacle avoidance Sonar	
2.2.5 Topside Interface and Vehicle Simulations	
2.2.6 In-Water Testing (Task 3.5)	
2.3 Acoustic Communications	
2.3.1 Summary	
2.3.2 Introduction	
2.3.3 Acoustic remote piloting and positioning	
2.3.4 High-speed acoustic communications	
References for Section 2.3	
2.4 Environmental Assessment and Modeling: PI: Dr. George V. Frisk	
2.4.1 Background	
2.4.1.1 Turbidity	
2.4.1.2 Turbidity Measurement	
2.4.1.3 Advantages of Turbidity Measurement	
2.4.1.4 Disadvantages of Turbidity Measurement	
2.4.2 Methodology	
2.4.2.1 Port Everglades Turbidity Measurement	
2.4.3 Results	

- 2.4.3.1 Temporal Dependence
- 2.4.3.2 Spatial Dependence
- 2.4.3.3 Turbidity Correlation
- 2.4.4 Conclusion / Discussion
- 2.4.5 Future Work
- References for Section 2.4
- 2.5 Development of a High Resolution Imaging Sonar for Underwater Inspections PI: Dr. Steven Schock
  - 2.5.1 Introduction
  - 2.5.2 Deliverables
    - 2.5.2.1 Acoustic, electric and analog design
    - 2.5.2.2 Data acquisition PCB fabrication
    - 2.5.2.3 Hydrophone and transmitter fabrication and performance
    - 2.5.2.4 Integration and testing of hydrophone segments, sonar processor and data acquisition system on the bench
    - 2.5.2.5 Fabrication of the hydrophone array and integrated electronics
    - 2.5.2.6 Bench and tank tests of the SLIS
- 2.6 Experimental determination of the hydrodynamic/dynamic characteristics of a small underwater vehicle for port security. PI: Karl von Ellenrieder
  - 2.6.1 Summary
  - 2.6.2 Introduction
    - 2.6.2-1 Background
  - 2.6.3 Experimental Setup
  - 2.6.4 Experimental Results
    - 2.6.4-1. Hydrodynamic Coefficients
    - 2.6.4-2. Variation of Thrust Output with Rudder Angle
  - 2.6.5 Conclusions
  - References for Section 2.6
- 2.7 Hydrodynamics and Dynamics Analyses of the Remotely-Piloted Unmanned Underwater Vehicle (RPUUV) PI: Dr. P. Ananthakrishnan
  - 2.7.1. Introduction
    - 2.7.1.1 Basic Vehicle Characteristics
    - 2.7.1.2 Vehicle Motion
    - 2.7.1.3 Hydrodynamics of the RPUUV
    - 2.7.1.4 Motion Simulations
    - 2.7.1.5 Contributions of the Project
  - 2.7.2. Estimation of Drag and Propeller Thrust and Power
    - 2.7.2.1 Six DOF Rigid Body Equations of Motion
    - 2.7.2.2 Horizontal Plane Three DOF Equations of Motion
    - 2.7.2.3 Vertical Plane Three DOF Equations of Motion
    - 2.7.2.4 Method of Analysis
  - 2.7.3. Determination of Hydrodynamic Forces and Moments

- 2.7.3.1 Unsteady Flow Hydrodynamics
- 2.7.3.2 Determination of Unsteady Hydrodynamic Forces and Moments
- 2.7.3.3 Green's Theorem and Boundary-Integral Method
- 2.7.3.4 Effect of Sea Bottom on Hydrodynamic Coefficients
- 2.7.3.5 Viscous Drag Force
- 2.7.3.6 Fin Force
- 2.7.3.7 Other External Forces and Moments
- 2.7.4. Simulation of Vehicle Motions: Discussions and Findings
  - 2.7.4.1 Horizontal Plane Motion With and Without Aft Fins
  - 2.7.4.2 Vertical Plane Motion With and Without Aft Fins
  - 2.7.4.3 Vertical Plane Motion With Mast and Modem
  - 2.7.4.4 RPUUV Motion Over Sea Bottom
- 2.7.5 Conclusion
- References for Section 2.7
- 2.8 RPUUV Navigation and Control: PI: Dr. Edgar An
  - 2.8.1 Summary
  - 2.8.2 Introduction
  - 2.8.3 Methods, Assumptions, and Procedures
    - 2.8.3.1 Navigation Sensor Considerations
    - 2.8.3.2 Navigation Performance Evaluation
    - 2.8.3.3 Review of an Error State Kalman Filter Without Gyros
    - 2.8.3.4 Review of an Error State Kalman Filter With Gyros
  - 2.8.4 Results and Discussion
  - 2.8.5 Conclusion
  - 2.8.6 Recommendations
  - 2.8.7 References
- 2.9 Chemical Sensors: PI: Dr. Richard Granata
  - 2.9.1 Summary
  - 2.9.2 Introduction
  - 2.9.3 Methods, Assumptions, and Procedures
    - 2.9.3.1 Primary Test Materials
    - 2.9.3.2 Primary Test Equipment
    - 2.9.3.3 Experiments
  - 2.9.4 Results and Discussion
  - 2.9.5 Conclusions and Recommendations
  - References for Section 2.9 - Chemical Detector
- 3.0 HIGH DEFINITION HIGH FRAME RATE COLOR CAMERA FOR SURVEILLANCE PI: Dr. William Glenn
  - 3.1 Summary
  - 3.2 Objective, Tasks, and Deliverables
  - 3.3 Results and Discussion
  - 3.4 Patents Filed During this Period
  - 3.5 Conclusion

### 3.6 Recommendation

## 4.0 STEREO AND MULTI-VIEW IMAGE AND VIDEO STABILIZATION, CALIBRATION, CODING, ANALYSIS AND PLAYBACK

PI: Dr. Borko Furht

### 4.1 Overview

### 4.2 Introduction

#### 4.2.1 Project Description

#### 4.2.2 Project Scope and Objectives

#### 4.2.3 Project Team

### 4.3 An Empirical Study on Video Stabilization

#### 4.3.1 Problem Statement

#### 4.3.2 Literature Survey

#### 4.3.3 Evaluation of Motion Estimation Methods

##### 4.3.3.1 Motion Models

###### 4.3.3.1.1 Translation

###### 4.3.3.1.2 Euclidean Transformation

###### 4.3.3.1.3 Similarity Transformation

###### 4.3.3.1.4 Affine Transformation

##### 4.3.3.2 Integral Projection

##### 4.3.3.3 Phase Correlation

##### 4.3.3.4 KLT-based feature tracking

##### 4.3.3.5 Handling Outliers in Motion Estimation by X84 Rule

##### 4.3.3.6 KLT-based block matching

##### 4.3.3.7 Test Results using Pairs of Images with Synthetic Motion

##### 4.3.3.8 Generating Synthetic Vibrations

##### 4.3.3.9 Measuring the Accuracy of Motion Estimation Methods

##### 4.3.3.10 Test Results using Videos with Synthetic Motion

##### 4.3.3.11 Reducing Error Accumulation

#### 4.3.4 Evaluation of Motion Correction Methods

##### 4.3.4.1 Motion vector integration

##### 4.3.4.2 Frame position smoothing

##### 4.3.4.3 Kalman filtering

##### 4.3.4.4 Test Results

#### 4.3.5 A Complete Algorithm for Video Stabilization

#### 4.3.6 Experimental Results

#### 4.3.7 Conclusion

#### References for Section 4.3

### 4.4 Computational Stereo: Camera Calibration

#### 4.4.1 Introduction

#### 4.4.2 Tsai's Camera Model

#### 4.4.3 A Flexible New Technique for Camera Calibration

#### 4.4.4 Camera Calibration Using the Caltech Toolbox

##### 4.4.4.1 System Requirements

##### 4.4.4.2 Getting Started

- 4.4.4.3 Calibration Example
- 4.4.4.4 About the Camera parameters
  - 4.4.4.4.1 Intrinsic parameters (camera model)
  - 4.4.4.4.2 Extrinsic parameters
- 4.4.5 Camera Calibration Using 3D Calibration Object
- References for Section 4.4
- 4.5 Stereo Correspondence
  - 4.5.1 Introduction
  - 4.5.2 Stereo Correspondence Algorithms
    - 4.5.2.1 Local Stereo Matching
    - 4.5.2.2 Global Stereo Matching
      - 4.5.2.2.1 Belief Propagation
      - 4.5.2.2.2 Graph Cut
      - 4.5.2.2.3 Other Stereo Matching Algorithms
    - 4.5.2.3 Evaluations
  - 4.5.3 Framework
    - 4.5.3.1. Arbitrarily-shaped windows
    - 4.5.3.2. Progressive edge-based stereo matching
  - 4.5.4 Experimental Design and Results
  - References for Section 4.5
- 4.6 3D and Multi-View Video Technologies
  - 4.6.1 Introduction
  - 4.6.2 Overview of Multi-View Video System
    - 4.6.2.1 Brief Overview of Binocular Vision
  - 4.6.3 Stereoscopic and Multi-View Video Player
    - 4.6.3.1 Player Architecture
    - 4.6.3.2 Stereoscopic Video Playback
  - 4.6.4 Experimental Methodology
    - 4.6.4.1 Quality Evaluation Tests
  - 4.6.5 Results and Discussion
    - 4.6.6.1 A New Hypercube Prediction Algorithm
    - 4.6.6.2 Experimental Setup
    - 4.6.6.3 Results and Discussion
  - References for Section 4.6
- 4.7 Object Segmentation Using Depth Information
  - 4.7.1 Introduction
  - 4.7.2 Background
    - 4.7.2.1 Vision science
    - 4.7.2.1 Stereo vision and the disparity map
  - 4.7.3 The Proposed Model
  - 4.7.4 Ongoing Work
  - 4.7.5 Conclusion
  - References for Section 4.7
- 4.8 Summary of Contributions and Deliverables
  - 4.8.1 Video Stabilization Contributions and Deliverables

- 4.8.2 Camera calibration and Stereo Correspondence Contributions and Deliverables
- 4.8.3 3D Video Coding and Playback Contributions and Deliverables
- 4.8.4 Object Segmentation Using Depth Information Contributions and Deliverables

## LIST OF FIGURES

### *Figures for Section 2.2*

- Figure 2.2.1: RPUUV with Modem Transducer & Electronics Integrated
- Figure 2.2.2: The RPUUV with High Resolution Sonar Payload
- Figure 2.2.3: Chemical Sensor Component Packaging
- Figure 2.2.4: The RPUUV with Chemical Sensor Payload
- Figure 2.2.5: Year 1 RPUUV Modifications
- Figure 2.2.6: LED Light Ring for the RPUUV
- Figure 2.2.7: Keller America Series 30x OEM Pressure Transducer
- Figure 2.2.8: Pressure Sensor Integrated in RPUUV Tail Section
- Figure 2.2.9: Pressure Sensor Calibration Results
- Figure 2.2.10: Pressure Sensor Error vs Depth
- Figure 2.2.11: Cruz-Pro OEM Depth Sounder
- Figure 2.2.12: Cruz-Pro Depth Sounder Installed in RPUUV Nose Section
- Figure 2.2.13: Tank Test Calibration Results for Cruz Pro Depth Sounder
- Figure 2.2.14: Obstacle Avoidance Sonar Packaging
- Figure 2.2.15: Obstacle Avoidance Sonar Beam Angles and Directions
- Figure 2.2.16: Obstacle Avoidance Sonar Transducers
- Figure 2.2.17: Obstacle Avoidance Sonar Nose-Cone and Electronics
- Figure 2.2.18: The Topside GUI Display Showing the Vehicle Track
- Figure 2.2.19: The Topside GUI Display Showing the Vehicle Depth & Bottom Location
- Figure 2.2.20: Operation Equipment for RPUUV Testing
- Figure 2.2.21: Photographs from Port Everglades Trials
- Figure 2.2.22: Photographs from November 2006 Port everglades testing
- Figure 2.2.23: Lake Testing with Surface Piercing Mast
- Figure 2.2.24: Testing of Year 2 RPUUV with Modem Control Only
- Figure 2.2.25: Marina Testing of RPUUV with Short Tow-Float Cable

### *Figures for Section 2.3*

- Figure 2.3.1: Overview of the RPUV control using a tow-float and acoustic waves.
- Figure 2.3.2: Detailed diagram of the RPUV control using acoustic waves.
- Figure 2.3.3: The remote control components of the RPUV.
- Figure 2.3.4: High-level flow chart of the acoustic piloting and positioning at the user end.
- Figure 2.3.5: High-level flow chart of the acoustic piloting and positioning at the RPUV end.
- Figure 2.3.6: Acoustic remote piloting electronics.



- Figure 2.3.7: USBL positioning array (left), coupled IMU and USBL Array (center) and XSens MTi IMU (right).
- Figure 2.3.8: Detailed system diagram of the acoustic piloting software and hardware.
- Figure 2.3.9: Remote acoustic piloting pool test.
- Figure 2.3.10: Acoustic piloting in SeaTech marina, pilot view.
- Figure 2.3.11: Aerial view of the FAU SeaTech marina.
- Figure 2.3.12: Frame transformation and motion compensation of the USBL system.
- Figure 2.3.13: Kalman filter for USBL motion compensation.
- Figure 2.3.14: USBL Mast mounted on a kayak.
- Figure 2.3.15: USBL test setup.
- Figure 2.3.16: HS-HFAM source (left) and receiver (right, courtesy of EdgeTech Inc.).
- Figure 2.3.17: Experiment overview, SeaTech Marina, Port Everglades, Florida.

*Figures for Section 2.4*

- Figure 2.4.1: Seapoint Turbidity Meter
- Figure 2.4.2: Cross Section of Seapoint Turbidity Meter Optical Components
- Figure 2.4.3: Port Everglades with Component Region
- Figure 2.4.4: Region 1 Turbidity Profiles
- Figure 2.5.5: Region 2 Turbidity Profiles
- Figure 2.4.6: Region 3 Turbidity Profiles
- Figure 2.4.7: Region 4 Turbidity Profiles
- Figure 2.4.8: Spatial Turbidity Profiles by Region
- Figure 2.4.9: Tide Height vs. Average Turbidity for Region 1
- Figure 2.4.10: Tide Height vs. Average Turbidity for Region 2
- Figure 2.4.11: Tide Height vs. Average Turbidity for Region 3
- Figure 2.4.12: Tide Height vs. Average Turbidity for Region 4

*Figures for Section 2.5*

- Figure 2.5.1 SLIS focusing on an object attached to a hull
- Figure 2.5.2 Signal processing flow chart for SLIS
- Figure 2.5.3 256 hydrophone elements attached to preamp and multiplexer PCBs
- Figure 2.5.4 The 256 channel hydrophone array and hemispherical projector
- Figure 2.5.5 Transmitting sensitivity of hemispherical projector
- Figure 2.5.6 The horizontal and vertical beam pattern functions
- Figure 2.5.7 The integrated sonar processor and data acquisition system
- Figure 2.5.8 The 256 channel hydrophone array integrated into electronics bottle
- Figure 2.5.9 SLIS image of concrete block generated during tank testing
- Figure 2.5.10 Acoustic image showing 135 degree field of view of the SLIS
- Figure 2.5.11 Simulated and measured resolution performance of the SLIS

*Figures for Section 2.6*

- Figure 2.6.1: Magnitude and direction of lift-thrust-power vs. thrust angle for a ducted rotor.
- Figure 2.6.2: The RPUUV Model.

- Figure 2.6.3: Definition of yaw angle, rudder angle and coordinate system.  
Figure 2.6.4: Experimental Setup.  
Figure 2.6.5: Hydrodynamic coefficients.  
Figure 2.6.6: Variation of thrust, yaw moment and thrust angle with rudder angle.  
Figure 2.6.7: PIV images of the flow along the articulated RPUUV tail section for the constant thrust case.

*Figures for Section 2.7*

- Figure 2.7.1.1 Illustrative sketch of the RPUUV  
Figure 2.7.2.1 Body-fixed coordinates and notations used in the RPUUV dynamics formulation  
Figure 2.7.3.1 Green's theorem for body motion in infinite fluid.  
Figure 2.7.3.2 Discretization of the body surface  
Figure 2.7.3.3 RPUUV motion above sea bottom: modeling using method of images.  
Figure 2.7.4.1 Trajectory of RPUUV without fins:  $T = 5$  [N], thruster angle  $\alpha = 10$  [deg] and  $x_G = +0.035$  [m]  
Figure 2.7.4.2 Trajectory of RPUUV with fins:  $T = 5$  [N], thruster angle  $\alpha = 10$  [deg] and  $x_G = +0.035$  [m], chord = span = 0.05 [m]; fin distance = 0.4 [m] aft.  
Figure 2.7.4.3 Trajectory of RPUUV with fins:  $T = 5$  [N], thruster angle  $\alpha = 40$  [deg] and  $x_G = +0.035$  [m], chord = span = 0.05 [m]; fin distance = 0.4 [m] aft.  
Figure 2.7.4.4 Trajectory of RPUUV (in the vertical plane) without fins:  $T = 5$  [N], thruster angle  $\alpha = 10$  [deg] and  $x_G = +0.035$  [m].  
Figure 2.7.4.5 Trajectory of RPUUV (in the vertical plane) without fins:  $T = 5$  [N], thruster angle  $\alpha = 10$  [deg] and  $x_G = +0.035$  [m], chord = span = 0.05 [m]; fin distance = 0.4 [m] aft.  
Figure 2.7.4.6 Trajectories of RPUUV with fins for thruster angle  $\beta = +40$  [deg] (solid line) and  $-40$  [deg] (dashed line)  
Figure 2.7.4.7 Trajectories of RPUUV with modem and mast and without fins for various thruster angles  
Figure 2.7.4.8 Effect of fins and their locations on the trajectories of RPUUV with modem and mast. Thrust = 5 [N] and thruster angle = 0 [deg].  
Figure 2.7.4.9 Vehicle motion without mast and fins but with modem for thrust = 5 [N] at thrust angles 0 [deg], 1 [deg] and 2 [deg].  
Figure 2.7.4.10 Horizontal plane motion in infinite fluid and over sea bottom ( $h=0.085$  m) both with thrust = 5 [N] and thrust angle = 10 [deg].  
Figure 2.7.4.11 Horizontal plane motion in infinite fluid and over sea bottom ( $h=0.085$  m) both with thrust = 5 [N] and thrust angle 10 [deg] and aft fixed fins.  
Figure 2.7.4.12 Temporal evolution of sway velocity with a perturbation introduced at  $t = 50$  [s]. The vehicle with aft fins is operating 0.085 [m] above ground with thrust = 5 [N] and thrust angle 10 [deg].

*Figures Section 2.8*

- Figure 2.8.1: Dead reckoning drift due to accelerometer bias (Case #1).  
Figure 2.8.2: Acc bias estimation performance (Case #1).  
Figure 2.8.3: Velocity estimation performance (Case #1).

Figure 2.8.4: Position estimation performance (Case #1).  
Figure 2.8.5: Position estimation performance with slower GPS fixes (Case #1).  
Figure 2.8.6: Velocity estimation performance with slower GPS fixes (Case #1).  
Figure 2.8.7: Gyro bias estimation performance (Case #2).  
Figure 2.8.8: Comparison of TCM2 measurements and their estimates (Case #2).  
Figure 2.8.9: Simulated acc bias estimation performance (Case #3A).  
Figure 2.8.10: Simulated gyro bias estimation performance (Case #3A).  
Figure 2.8.11: Simulated position estimation performance (Case #3A).  
Figure 2.8.12: Simulated velocity estimation performance (Case #3A).  
Figure 2.8.13: Simulated velocity estimation performance (Case #3B).  
Figure 2.8.14: Simulated position estimation performance (Case #3B).  
Figure 2.8.15: Simulated velocity estimation performance (Case #3C).  
Figure 2.8.16: Simulated position estimation performance (Case #3C).  
Figure 2.8.17: Simulated acc bias estimation performance (Case #3D).  
Figure 2.8.18: Simulated gyro bias estimation performance (Case #3D).  
Figure 2.8.19: Simulated position estimation performance (Case #3D).  
Figure 2.8.20: Simulated velocity estimation performance (Case #3D).  
Figure 2.8.21: Simulated velocity estimation performance (Case #4).  
Figure 2.8.22: Simulated position estimation performance (Case #4).  
Figure 2.8.23: Simulated USBL measurement noise characteristics (Case #4).

*Figures for Section 2.9*

Figure 2.9.1: WETStar Fluorometer.  
Figure 2.9.2: Proposed design schematic no. 1  
Figure 2.9.3: Proposed design schematic no. 2

*Figures for Section 3*

Figure 3.1 Pair of HDMAX Cameras  
Figure 3.2 Two Display Interfaces with Control Computer

*Figures for Section 4*

Figure 4.1. A generic video surveillance system: block diagram.  
Figure 4.2. Block diagram of the proposed foreground segmentation system.  
Figure 4.3. Movement related change detection.  
Figure 4.4. "Stationary" change detection.  
Figure 4.5. Change detection plot.  
Figure 4.6. F-H algorithm: (a) the original frame, (b) the F-H output ( $k = 100$ ) and without the post-processing step, and (c) the F-H output after post-processing that shows the undesired effect of losing the small boat.  
Figure 4.7. Reducing the number of false positives with modified post-processing based on color moments and edge bins: (a) F-H segmented image without modified post-processing (79 segments), and (b) with our modified post-processing (15 segments).

Figure 4.8. Representative frame: (a) original frame, (b) reference background frame, (c) F-H segmentation, (d) final segmentation result.

Figure 4.9. Object Classification in a Visual Surveillance System.

Figure 4.10 Components and work flow of object classification.

Figure 4.11 Detecting ships by background subtraction and double thresholding with hysteresis.

Figure 4.12 Examples of Express Cruise.

Figure 4.13 Examples of Motor Yacht.

Figure 4.14 Examples of Recreational.

Figure 4.15 Examples of Speed Boat.

Figure 4.16 Examples of Sportfish.

Figure 4.17 Examples of Water Taxi

Figure 4.18 Demonstration of k Nearest Neighbor classification algorithm.

## LIST OF TABLES

Table 2.3.1. HS-HFAM Data Packet Specifications

Table 2.3.2. HS-HFAM Performance Summary at 88 m

Table 2.7.3.1 Principal Geometry and Hydrodynamics Related Quantities

Table 2.7.4.1 Unsteady hydrodynamic coefficients for various altitudes above sea bottom.

Table 2.8.1: A comparison of existing IMU product specifications

Table 2.8.2: A comparison of existing IMU product specifications

Table 2.8.3: A comparison of existing IMU product specifications

Table 4.1 An example of calculating a RMSE value between two pixels based on the intensity values of the square windows of size 3

Table 4.2 An overview of stereo matching algorithms with top performance on the Middlebury stereo data

Table 4.3 The performance of using different extents of arbitrarily-shaped windows (in terms of percentage bad pixels for non-occluded regions and all regions)

Table 4.4 The parameter settings for the four stereo data

Table 4.5 Improvement of using progressive edge-based stereo matching over without using edge-based strategy (in terms of percentage of bad pixels for non-occluded, all and disparity discontinuity regions, with threshold of 1)

Table 4.6 An evaluation of our algorithm on the Middlebury data (in terms of percentage of bad pixels for non-occluded, all, and disparity discontinuity regions; the subscripts of the results are our rankings amongst other state-of-the-art algorithms, with thresholds 1 and 0.5)

Table 4.7 Reference views for an eight camera array

## EXECUTIVE SUMMARY

The Center for Coastline Security Technology (CCST) focuses on research, simulation, and evaluation of coastal defense and marine domain awareness equipment, sensors, and components. It builds upon the existing efforts and expertise in coastal systems and sensor research at the Institute for Ocean and Systems Engineering (IOSE), the Imaging Technology Center, the Department of Computer Science and Engineering, and the University Consortium for Intermodal Transportation Safety and Security at Florida Atlantic University.

New technologies are needed to enhance surveillance and inspections of marine activities in the coastal zone that includes major ports, small inlets, beaches, remote coastal areas, and their approaches. The task is to effectively integrate sensors with underwater, surface, and airborne autonomous and remotely-operated platforms and to incorporate video and image analysis and data mining methods to quickly and effectively identify threat events.

The technologies that will be developed in this program are:

- 1) *Underwater vehicles for survey and inspection:* In the CCST program a low cost, one man operated, remotely piloted unmanned, untethered, underwater vehicle, is being developed which will provide real time underwater video and sonar images to a topside console. The specific application to be addressed is underwater inspection by rapid response teams, and routine inspection activities, currently carried out by scuba divers. This technology is intended to reduce the need for divers on a 24/7 basis. During year one of the program a vehicle was developed with a tow float and a RF antenna to provide the underwater video and sonar data to a topside console. In year two a tetherless capability has been added by replacing the tow float with a high speed acoustic modem. In addition a high resolution sonar system has been developed which will be mounted on the vehicle in the third year of the program. The high resolution sonar will operate in side scan mode and will rotate about it's axis to provide images from different aspects. The design of the sonar is an important first step towards the overall objective of developing high resolution underwater images of ship hulls and port seawalls.
- 2) *High Definition Video Systems:* High-definition video cameras provide an order of magnitude improvement in field of view and/or range over those achievable with conventional video systems. They are thus a necessity for harbor surveillance; however, their implementation in this environment is limited by size and cost. At Florida Atlantic University's Imaging Technology Center, a compact super-high-definition camera (with four times the resolution of conventional high-definition video cameras) has been developed and is ready for the commercial market, the primary customers being the cinematic film industry. For the port security application there are several research issues being addressed under this program, specifically, recording the output of the camera, managing the high-data-rate output of the camera, testing the camera in the marine environment, and combining a pair of the cameras with a matched pair of digital video projectors for real-time 3D surveillance. The test and evaluation issue will be addressed by the ITC in collaboration with NAVSEA Carderock's South Florida Test Facility, which has towers overlooking Port Everglades, and the adjacent inlet, which are already

used by the USCG for video surveillance. Software enhancement of 3D imaging using the HDMAX camera will be addressed by Florida Atlantic University's Department of Computer Science and Engineering.

This document is the final report for year two of this three-year program and describes the progress on the following projects

- The Development of a Remotely Piloted, Unmanned, Untethered, Underwater Vehicle (RPUUV),
- HDMAX High-Resolution QUAD HD Progressive Scan Electronic Camera System,
- 3D Imaging and 3D Video Technologies for Coastline Security Applications

The project includes the activities of ten principle investigators. The following provides a summary of the achievements of each element of the program.

### **Development of a Remotely Piloted Unmanned Underwater Vehicle** **PI: Dr. Stewart Glegg, Project Manager: Robert Coulson**

The development of the Remotely Piloted Unmanned Underwater Vehicle is described in Section 2.2. The objective of year two of this program was to develop a vehicle that is controlled by a topside console through an acoustic link, and to enhance vehicle performance with a suite of different sensors.

The vehicle that has been developed features a vectored thruster with an 80 deg angular range, which allows the vehicle to maneuver in tight spaces. The weight of the vehicle is approximately 35 lbs and it is easily launched and recovered by a single operator from the side of a small vessel. The vehicle includes an onboard computer which processes the sensor data, the underwater video and the output from an onboard compass, pitch and roll sensor. In the vehicle developed in year one of this program, the data from these systems is relayed through a wireless RF link on the tow float to the topside console using a remote desktop capability. The vehicle is controlled through the RF link using a commercially available remote control device developed for model aircraft. In the second generation vehicle, developed in year two, control is achieved through an underwater acoustic link.

A complete description of the vehicle modifications and in water tests which took place during year two is given in Section 2.2. Also included is a description of the in-water test, which was carried out in April 2007, of the second generation vehicle which was controlled using an acoustic link. The major achievement of the program in year two of this project is that acoustic modem control of the vehicle was demonstrated in a shallow water marina, providing successful control of the vehicle over a range of ~75m in a cluttered environment. The vehicle was sufficiently controllable that it could be brought alongside and recovered using acoustic communications to control the vectored thruster. To our knowledge this is the first time that an underwater vehicle has been controlled in real time through an acoustic communications device.

## **Acoustic Communications**

**PI: Dr. P. Beaujean**

The main objective of this portion of the project is to develop communication systems for the purpose of transmitting and receiving information wirelessly between a user and the Remotely Piloted Untethered Underwater Vehicle (RPUUV). Transmitted information is used to pilot the RPUUV and relay its position. Information received from the RPUUV combines acoustic images of the environment and status report of the vehicle. During the first year of this project radio wave (WiFi) communication was used to control the vehicle. Whenever the tow-float solution becomes impractical, a slower but fully wireless acoustic modem is to be used. The design must consider the issues associated with acoustic communications in port at high data rates, using a high-frequency acoustic modem, and the piloting and tracking of the RPUUV, using a command-and-control acoustic modem. During year two of the program a vehicle control system using an acoustic link has been developed, installed on the vehicle and tested. The test results showed that the vehicle was easily controlled using this technology.

## **Environmental Assessment and Modeling: Monitoring Turbidity in Ports**

**PI: Dr. George V. Frisk**

The overall goal of this project is to characterize Port Everglades both acoustically and optically as these properties relate to the operation of the Remotely Piloted Unmanned Underwater Vehicles (RPUUV) technology being developed for the Center for Coastline Security Technology (CCST). Once the relation of these properties to the functionality of the RPUUV sonar and video systems is adequately understood, this approach can be applied to other port environments in which a similar surveillance system may be employed.

The specific objectives for year 2 of the project were to develop a methodology and system for monitoring the turbidity levels in the Port Everglades environment, including the identification of a COTS optical system for measuring the temporal and spatial variability of turbidity levels. For this purpose a Seapoint Turbidity Meter has been chosen and integrated with a Falmouth Scientific Conductivity, Temperature, and Depth (CTD) instrument for simultaneous water measurements of salinity, temperature, and sound speed, in addition to turbidity. A methodology for the deployment of the device has been chosen to assure a minimization of error and consistency in the data. Using this methodology aboard an FAU Ocean Engineering research vessel, 15 at-sea trips and more than 180 profiles have been collected and analyzed. These measurements have shown a high degree of variability within the Port on a temporal and spatial basis ranging from between 1 and 10 Nephelometric Turbidity Units (NTU). Identification of the suitability of areas around the port to the operation of devices that rely on optical clarity can be recognized by the separation of the port into specific regions exhibiting similar turbidity characteristics. As expected, temporal variations showed a high correlation to tidal height; however, no relation was found between turbidity and current, salinity, or rainfall. Future work includes detailed spectral absorption and attenuation measurements to gather information on the constituents contributing to the underwater optical degradation.

## **Development of a High Resolution Imaging Sonar for Underwater Inspections**

**PI: Dr. Steven Schock**

A SLIS (side looking image sonar) was designed, fabricated and tested to establish the feasibility of generating very high resolution images with a wide field of view ( $>90$  degrees) using a single transmission. Widely used commercial acoustic cameras are not practical for UUVs conducting harbor searches because those cameras 1) use several transmissions to form an image requiring a very stable platform and motion compensation and 2) have a narrow field of view (29 degrees) which limits search rates. The results of SLIS tank tests validated the array design and signal processing concepts that allow generation of wide field of view images using a single transmission. Simulations and measurements of range and azimuthal resolution for targets in the near field of the array closely agreed, thereby, showing that the azimuthal resolution of one acoustic wavelength (1.5 mm at 1 MHz) is achievable out to ranges of one array length. The measured field of view of the SLIS was 135 degrees at 1.1 MHz which is a substantial improvement over commercial acoustic cameras with a field of view of only 29 degrees.

## **Experimental Determination of the Hydrodynamic/dynamic Characteristics of a Small Underwater Vehicle for Port Security**

**PI: Karl von Ellenrieder**

The objectives of this research were to study the hydrodynamic design and dynamic response of the RPUUV. An experimental model, which allows for reconfiguration of the vectored-thruster propulsion system (the control surface of the vehicle) was developed and tested in various roll, pitch and yaw configurations in order to determine the hydrodynamic coefficients and thrust output of the vehicle.

Force/torque and particle image velocimetry measurements were conducted in a water flume/towing tank to: 1) determine the hydrodynamic drag, lift and moment coefficients acting on the vehicle hull for zero rudder angle and yaw angles up to thirty degrees, and 2) measure the magnitude and direction of the thrust produced with the vehicle at a yaw angle of zero degrees and rudder deflection angles of up to thirty degrees.

The measured drag coefficient was very close to that predicted by theory. It was found that the magnitude of the thrust vector varies nonlinearly with rudder angle and for nonzero rudder angles the thrust vector does not point in the same direction as the thruster. PIV images reveal that at rudder deflection angles of twenty five and thirty degrees the flow upstream of the propeller inlet has separated from the tail section and impinges at a large angle to the tail, thereby reducing both the thrust deflection angle as well as the total yaw moment acting on the vehicle. The experimental data are expected to be useful for predicting the open loop response of vehicles in the field and for the development of a closed loop control system for the RPUUV.



## **Hydrodynamics Analysis and Simulations for Design and Operation of a Remotely-Piloted Unmanned Underwater Vehicle (RPUUV)**

**PI: Dr. P. Ananthakrishnan**

The Year 2 objective of the project was to carry out hydrodynamic and dynamic analyses and simulations of the remotely-piloted unmanned underwater vehicle and based on the results and findings contribute to improvements in design and performance of the vehicle. The problem formulations, solution methods, simulations, new findings and contributions are presented.

Section 2.7 of this report describes a boundary-integral algorithm based on the Green's theorem that has been developed to determine the unsteady hydrodynamic coefficients of the vehicle. Sea bottom effects are modeled based on the method of images. Results show that the hydrodynamic coefficients are only significantly affected if the vehicle is very close to the bottom. Lift and drag forces on the vehicle, appendages and fins are modeled using experimentally-determined lift and drag coefficients.

Equations governing rigid-body vehicle motion, formulated using a body-fixed frame of reference, are integrated in time using Euler's scheme to simulate vehicle dynamics. Simulations were carried out for a range of scenarios and parameter values. The vehicle, without modem and mast, is found to be dynamically robust even without any fins. The addition of an appendage such as the modem transducer induces a pitch motion which can be easily controlled using the vectored thruster. Addition of a mast however induces a large unsteady pitch motion which is difficult to control either with thruster or any fixed fins. Plausible solutions to suppressing the mast-induced motions are (i) introducing a counter mast on the bottom or (ii) moving the center of gravity of the vehicle through a large distance forward; both solution are however not practical.

Dynamics of the vehicle is not affected significantly by the sea bottom even when the vehicle is very close to the bottom. The only limitation to the vehicle motion is caused by the actual bottom itself and not by the hydrodynamics aspect of the bottom.

## **RPUUV Navigation and Control**

**PI: Dr. Edgar An**

To monitor coastline security during a mission, the RPUUV operators must not only analyze in real-time the video and acoustic data via the high-speed acoustic link, but also position the vehicle accurately by directly controlling the vectored thruster. The latter task generally requires a great deal of effort from the operators, and thus it is highly desirable to automate the control process so that the operators can focus mostly on the data analysis and threat identification. One way to achieving this objective is to allow the operators to command using only waypoints or set points instead of controlling the vectored thruster's angle and speed. The vehicle must then be capable of determining its position and attitudes accurately, and self-adjusting the thruster dynamics accordingly. Currently, there is no navigation hardware / software on the RPUUV although the vehicle is capable of receiving its USBL position fixes but at a very slow update

rate. Controlling the position and attitude of the RPUUV adequately would require a much higher update rate, on the order of 10Hz. The main objective of the proposed task is to evaluate a number of inexpensive, alternative navigation sub-systems for the remotely piloted UUV. The Year 2 achievements consist of: researching the latest navigation sensors available on the market, investigating two navigation solutions suitable for the RPUUV, and evaluating the position error performance based on 3D vehicle motion simulation and at-sea data collected using the OEX AUV from FAU.

### **Chemical Sensors**

**PI: Dr. Richard Granata**

This section describes the formulation of a chemical method to detect underwater trace explosives, as well as the design of a field-deployable device to implement the chemical method. The research goals are identified, the primary test materials, equipment and experiments are described and the results are discussed. The chemical compound, europium thenoyltrifluoroacetone, has been identified as an integral part of a viable underwater chemical detection method for underwater explosive traces. The method uses a photoluminescent response of the europium compound with nitro-based explosives such as nitroglycerine. Feasibility of the method for use in seawater has been demonstrated to an estimated detection limit of 28 ppb using COTS components in a flow-through configuration. A report describing the details has been completed. Installation of the components in the UUV is in-progress to be followed by field testing.

### **High Definition High Frame-Rate Color Camera for Surveillance**

**PI: Dr. William E. Glenn**

The overall objective of this segment of the project is to develop a high-definition, high-frame-rate color video camera system for surveillance. During the first year of the program a 3840x2160 30P (30 FPS progressive scan) super-high-definition color CMOS camera—the HDMAX camera—with variable frame rate and remotely controlled infrared filter changer was designed, fabricated, tested, and demonstrated. This camera gathers 50 times the amount of information in its field of view as do standard-resolution video cameras often used for surveillance purposes. A flash-memory-based solid-state device for recording large amounts of image data generated by the camera was also designed, fabricated, and tested. Field tests demonstrated that the camera's high resolution makes it possible to do electronic zoom on sections of an image without permanent loss of the remaining portions of the field of view, and the high frame rate allows the use of moving target indication, velocity measurement, and the observation of brief events that help classify targets of interest. During the year covered by this report two upgraded HDMAX camera systems were built for use in next year's program in the investigation of 3D imaging, and a prototype video compression system was built and tested. Providing in excess of a 10:1 compression of video information without significant loss of useful information, this compression system, when combined with a solid-state recorder module, will allow nearly three hours of recording time per module. The HDMAX camera, video signal compressor, and solid-state recorder are ideally suited for video surveillance on ships, submarines, harbors, AUVs, and drone aircraft.

## **Stereo and Multi-View Image and Video Stabilization, Calibration, Coding, Analysis and Playback**

**PI: Dr. Borko Furht**

This report reviews the second year of research activities in the field of image and video analysis algorithms for coastline security. Our research work in the second year has been focused on developing robust techniques and methodologies for multi-view video capturing, analysis, delivery and presentation. This work extends our efforts from the first year which mainly focused on developing algorithms and techniques for motion detection, object tracking, and object classification in maritime scenes from single-view images and video sequences. As a set of deliverables of the second year research we proposed and implemented robust algorithms for compensation of camera vibration, 3D reconstruction from multiple images, 3D video player for playback, algorithms for multi-view and 3D video compression and image and video object segmentation algorithms using depth information.

## **1.0 INTRODUCTION**

### **1.1 Overview**

#### **1.1.1 Background**

The Center for Coastline Security Technology (CCST) focuses on research, simulation, and evaluation of coastal defense and marine domain awareness equipment, sensors, and components. It builds upon the existing efforts and expertise in coastal systems and sensor research at the Institute for Ocean and Systems Engineering (IOSE), the Imaging Technology Center, the Department of Computer Science and Engineering, and the University Consortium for Intermodal Transportation Safety and Security at Florida Atlantic University.

New technologies are needed to enhance surveillance and inspections of marine activities in the coastal zone that includes major ports, small inlets, beaches, remote coastal areas, and their approaches. The task is to effectively integrate sensors with underwater, surface, and airborne autonomous and remotely operated platforms and to incorporate video and image analysis and data mining methods to quickly and effectively identify threat events.

This effort includes activities at Florida Atlantic University's SeaTech campus, allowing researchers to leverage the existing U.S. Navy marine test & evaluation facilities, geographically combined with the adjacent major seaport at Port Everglades. This provides a unique land and aquatic test bed. Initial studies have focused on acoustic sensors and high definition underwater and surface video sensors mounted on unmanned fixed or mobile platforms. Emphasis has also been given to the development of optimal platforms for the efficient collection and integration of the information from multiple sensors.

#### **1.1.2 Technical Objectives**

As time progresses it is becoming increasingly apparent that the country's ability to provide elevated homeland security in ports and harbors is limited by operational costs. Budgets for port security are several times larger than they were before the events of 9/11/01 and cost is now a major issue for both federal and local agencies. Furthermore, when Navy ships dock in areas also used for civilian activities, security issues are more complex and require close collaboration between all agencies involved. The same principles apply in overseas ports, as evidenced by the attack on the USS Cole, and port security technology, which is portable to international locations, has an important role in force protection.

Given these prerequisites it is the primary objective of this program to develop new technology for port security that provides unique capabilities for security inspections, threat detection and rapid response, at lower operational costs. To achieve this objective, attention will be focused on the technologies in which the members of the Center have existing expertise, with the intent of turning these technologies into operational systems in a three year program.

The technologies that will be developed in this program are:

- 1) *Underwater vehicles for survey and inspection:* In the CCST program a low cost, one man operated, remotely piloted unmanned, untethered, underwater vehicle, is being developed which will provide real time underwater video and sonar images to a topside console. The specific application to be addressed is underwater inspections by rapid response teams, and routine inspection activities, currently carried out by scuba divers. This technology is intended to reduce the need for divers on a 24/7 basis. During year one of the program a vehicle was developed with a tow float and a RF antenna to provide the underwater video and sonar data to a topside console. In year two a tetherless capability has been added by replacing the tow float with a high speed acoustic modem. In addition a high resolution sonar system has been developed which will be mounted on the vehicle in the third year of the program. The high resolution sonar will operate in side scan mode and will rotate about it's axis to provide images from different aspects. The design of the sonar is an important first step towards the overall objective of developing a high resolution underwater images of ship hulls and port seawalls.
- 2) *High Definition Video Systems:* High definition video cameras provide an order of magnitude improvement in field of view and/or range over those achievable with conventional video systems. They are thus a necessity for harbor surveillance, but their implementation in this environment is limited by size and cost. At Florida Atlantic University's Imaging Technology Center, a compact super-high-definition camera (with four times the resolution of conventional high-definition video cameras) has been developed and is ready for the commercial market, the primary customers being the film industry. For the port security application there are several research issues being addressed under this program, specifically, recording the output of the camera, managing the high data output rate of the camera, testing the camera in the marine environment, and combining a pair of the cameras with a matched pair of digital video projectors for real-time 3D surveillance. The test and evaluation issue will be addressed by the ITC in collaboration with NAVSEA Carderock's South Florida Test Facility, which has towers overlooking Port Everglades, and the adjacent inlet, which are already used by the USCG for video surveillance. Software enhancement of 3D imaging using the HDMAX camera will be addressed by Florida Atlantic University's Department of Computer Science and Engineering.

In this report the details for year two of this program will be presented. The following projects are described

- The Remotely Piloted, Unmanned, Untethered, Underwater Vehicle (RPUUV),  
PIs Dr. S. Glegg
- Acoustic Piloting, Communications and Positioning  
PI: Dr. P.Beaujean
- Environmental Assessment and Modeling: Monitoring Turbidity in Ports  
PI: Dr. George V. Frisk
- Development of a High Resolution Imaging Sonar for Underwater Inspections  
PI: Dr. Steven Schock
- Experimental determination of the hydrodynamic/dynamic characteristics of a small underwater vehicle for port security

PI: Dr. von Ellenrieder

- Hydrodynamic and Dynamic Investigations for the Development of a Small Underwater Vehicle for Underwater Hull Inspection and Harbor Survey

PI: P. Ananthakrishnan

- RP UUV Navigation and Control

PI: Dr. Edgar An

- Chemical Sensors

PI: Dr. Richard Granata

- HDMAX High-Resolution QUAD HD Progressive Scan Electronic Camera System,

PI: Dr. W. Glenn,

- 3D Imaging and 3D Video Technologies for Coastline Security Applications

PI: Dr. B. Furht

## **2.0 The Remotely Piloted, Unmanned, Untethered, Underwater Vehicle (RPUUV)**

### **2.1 Background**

Currently unmanned underwater vehicles fall into two distinct classes: (1) Remotely operated vehicles that are tethered to a topside operations console. These devices are usually cage like and are designed to have a hovering capability for close up inspection of a site. They are limited by the necessity to drag a tether with them and so are rarely used for large area rapid surveys. (2) Autonomous Underwater Vehicles that are untethered and used extensively to carry out large area surveys for MCM applications. These vehicles are given a pre specified set of tasks, and have an on board navigation and object recognition capability. They are limited because currently they do not provide real time video or sonar images to the topside operator and rely on either on board intelligence or post deployment analysis to detect and evaluate targets. The autonomous capability requires sophisticated on board sensors that drive up the cost.

An alternative approach combines the remotely piloted features of an ROV with the advantages of the untethered AUV. To achieve this requires a high speed wireless underwater communications capability that is only just beginning to become available. It was proposed to develop this type of vehicle and the enabling technology as part of this program and the details of the vehicle development and test program will be described in section 2.2.

At the present time underwater wireless communication is carried out acoustically by use of an acoustic modem. Florida Atlantic University has had a strong program on acoustic modem research for the past ten years, and has developed a number of acoustic communication devices that are used on fully operational AUVs. However to achieve remotely piloted vehicle operation it was not clear at the beginning of this project that acoustic devices will be able to provide the communication rates needed, especially in a shallow water harbor environment. Progress on the development of this technology is described in section 2.3, including a description of how the technology has been installed onto the RPUUV.

To develop the on board sensor systems required, the topside console, and understand the data rate issues for an RPUUV, a vehicle was developed in year one of this program which used a short tether to a surface float with an RF antenna. The antenna has allowed for real time underwater video, navigation and sonar data to be transmitted back to the topside console in real time. Further development and in water testing of this system is described in section 2.2. Also included in Section 2.2 is a description of the second generation of RPUUV which is controlled through an acoustic modem. The major achievement of the program in year two of this project is that acoustic modem control of the vehicle was demonstrated in a shallow water marina, providing successful control of the vehicle over a range of ~75m in a cluttered environment. The vehicle was sufficiently controllable that it could be brought alongside and recovered using acoustic communications to control the thruster.

For an RPUUV to achieve its full potential an on board suite of sensors will be required which include side scan sonars, integrated obstacle avoidance sonars, chemical sensors, and an on board navigation capability. In the second year of this program we have continued studies on each of these sensor packages, with the intent of integrating them into an RPUUV in year three. These systems are described in section 2.5, 2.8 and 2.9. In addition good hydrodynamics is required for stability and lower power consumption and research in these areas are described in sections 2.6 and 2.7.

The operation of the communication system and other sensors will also depend on the details of the port environment, including speed of sound profiles, turbidity profiles and the local currents. A study to investigate these features in Port Everglades is described in section 2.4.



## **2.2 Development of a Remotely Piloted Unmanned Underwater Vehicle**

**PI: Dr. Stewart Glegg, Project Manager: Robert Coulson**

### **Tasks 3.1-3.6**

#### **2.2.1 Summary**

The development of the Remotely Piloted Unmanned Underwater Vehicle is described in Section 2.2. The objective of year two of this program was to develop a vehicle that is controlled by a topside console through an acoustic link, and to enhance vehicle performance with a suite of different sensors.

The vehicle that has been developed features a vectored thruster with an 80 deg angular range, which allows the vehicle to maneuver in tight spaces. The weight of the vehicle is approximately 35 lbs and it is easily launched and recovered by a single operator from the side of a small vessel. The vehicle includes an onboard computer which processes the sensor data, the underwater video and the output from an onboard compass, pitch and roll sensor. In the vehicle developed in year one of this program, the data from these systems is relayed through a wireless RF link on the tow float to the topside console using a remote desktop capability. The vehicle is controlled through the RF link using a commercially available remote control device developed for model aircraft. In the second generation vehicle, developed in year two, control is achieved through an underwater acoustic link.

A complete description of the vehicle modifications and in water tests which took place during year two is given in Section 2.2. Also included is a description of the in water test, which was carried out in April 2007, of the second generation vehicle which was controlled using an acoustic link. The major achievement of the program in year two of this project is that acoustic modem control of the vehicle was demonstrated in a shallow water marina, providing successful control of the vehicle over a range of ~75m in a cluttered environment. The vehicle was sufficiently controllable that it could be brought alongside and recovered using acoustic communications to control the vectored thruster. To our knowledge this is the first time that an underwater vehicle has been controlled in real time through an acoustic communications device.

#### **2.2.2 Introduction**

The main objective of this task is to develop the platform that will support the sensors being developed in the other parts of the project. During the first year of the program a vehicle was developed which has enabled further developments of sensors and communication systems. The first generation vehicle was attached to a tow float with an RF antenna through which the vehicle communicates with a topside console. This system has provided valuable information on the operational requirements for vehicle deployment, and, during year two has been modified significantly and tested in both open ocean and port environments. The major task for year two of the program was to build a

second generation vehicle, and to develop the designs for the attachment of the high resolution sonar, and the chemical sensor. The second generation vehicle differs from the first generation because it is controlled through an acoustic link which introduces a number of additional challenges. The development and in water testing of this vehicle is described in the following sections.

### **2.2.3 Re-design of the RPUUV to Support an Acoustic Modem, High Resolution Imaging Sonar & Explosives Detection Payloads (Tasks 3.1, 3.2, & 3.3)**

In year one of this work, the RPUUV was controlled solely by the Radio Communications (RC) link between the operator and a tow-float wired directly to the vehicle. Sensor data was similarly relayed via a WiFi communications channel. In year two a low frequency acoustic modem link was inserted to provide an alternate wireless command and control channel, and it is proposed in year three that a high speed, high frequency acoustic modem be added to allow for periodic wireless transmission of sensor and vehicle feedback data to the topside operator.

Detailed information regarding the design of these modems can be found in section 2.3 of this report. In section 2.2.3.1 we address their physical integration into the RPUUV platform. In water testing of the RPUUV equipped with the low frequency command and control modem will be discussed in section 2.2.7.

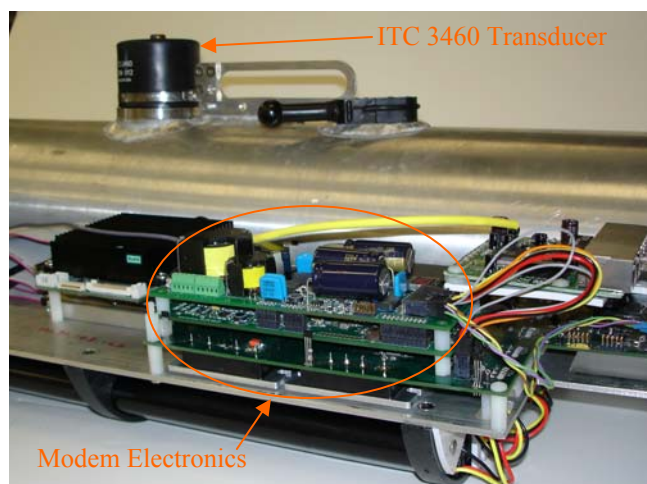
Two sensor payloads are being developed at FAU to be carried by the RPUUV. The data produced by these sensors is anticipated to be relayed via the high speed acoustic modem in the year three effort. The first payload is a high resolution imaging sonar being built by Dr. Steven Schock, the details of which are reported in section 2.5. The second payload is an explosives chemical detection package that is described in section 2.6, and is being built by Dr. Richard Granata. Preliminary design of the packaging and interfacing of these systems with the RPUUV has been completed and the details are presented in section 2.2.3.2.

#### **2.2.3.1 Acoustic Modem Integration**

The RPUUV motherboard that was developed in year one is designed to physically accommodate the acoustic modem electronics. This motherboard supplies the necessary voltage to run the signal processing and signal conditioning boards. The modems are interfaced with the RPUUV through the RS485 serial communications bus, and can also be accessed through the on-board Ethernet hub.

The hull of the RPUUV was already fitted with a welded flange during year one, on which the ITC3460 modem transducer could be mounted and wired into the modem electronics inside the vehicle.

Photographs of the RPUUV midsection showing the transducer mounted to the hull and the packaged electronics can be seen in Figure 2.2.1.

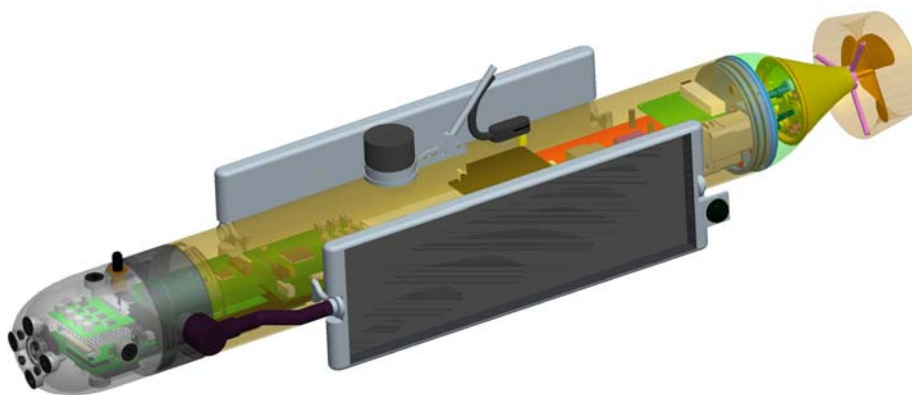


**Figure 2.2.1:** RPUUV with Modem Transducer & Electronics Integrated

In this configuration, the RPUUV operator still uses the standard RC controller handset, only now the receiver unit is included in the topside modem electronics. The modulated signals from the RC controller are digitized by the same tow-float electronics as before and this command is then sent via the modem instead of down the tow-float cable. In-water tests were conducted in the marina adjacent to FAU's Seatech campus and are discussed in section 2.2.7.

### 2.2.3.2 Payload Integration Design

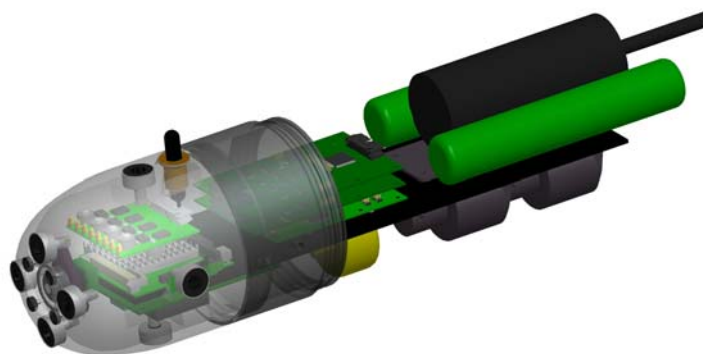
The first of two payloads that are being developed for the RPUUV is a High Resolution Imaging Sonar. This system has been designed and is undergoing initial testing at FAU. The details of this sonar are presented in section 2.5. Modeling has also been performed to establish how this system will be integrated with the RPUUV and a CAD model representation of this design is shown in Figure 2.2.2. CAD models of all the RPUUV versions and payloads are also available on a CD in Pro Engineering format.



**Figure 2.2.2:** The RPUUV with High Resolution Sonar Payload

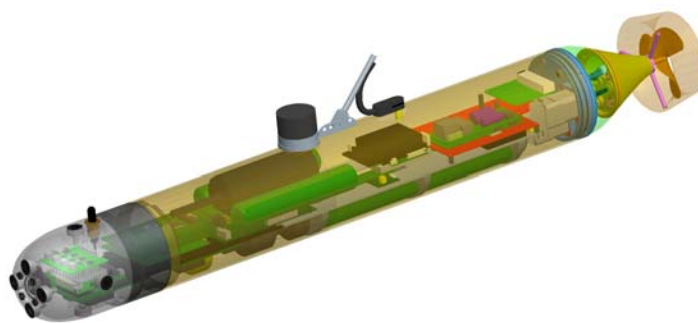
The high resolution sonar consists of slab-like arrays that will be mounted to a lengthened parallel mid-section of the RPUUV. An adapter ring will also be added to provide a unique interface for this sonar package allowing for penetrations through the hull that will carry the large number of conductors that are necessary to power the arrays and projectors as well as bring in their data to be processed inside the vehicle. The arrays will be mounted on pivots that will allow them to be tilted up to 45 degrees in an upward or downward look direction.

The second RPUUV payload under development at FAU is designed to detect explosive chemicals. The details of this sensor are presented in section 2.6. Some very preliminary modeling of the major components has been performed however and this low-level packaging detail can be seen in Figure 2.2.3. The fully integrated sensor package is tentatively modeled in Figure 2.2.4.



**Figure 2.2.3:** Chemical Sensor Component Packaging

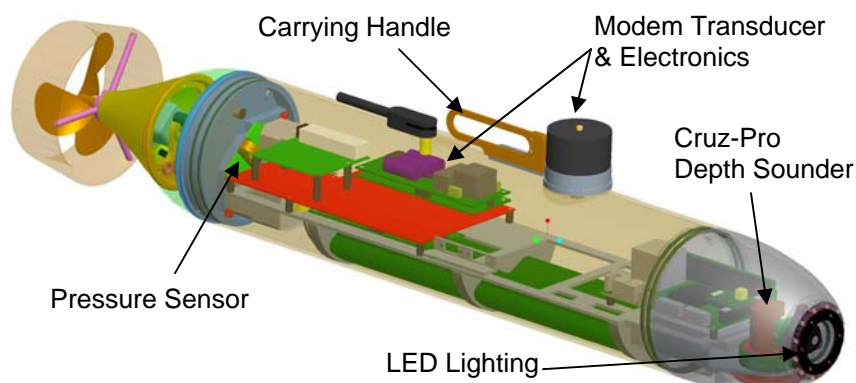
The chemical detection payload can be housed in the same parallel mid-section as the high resolution imaging sonar, but will have its own unique adapter ring that will be ported to allow for seawater intakes and expulsion via two micro-pumps. Accumulator style reagent reservoirs will replace used reagents with saltwater so that the overall buoyancy of the vehicle does not change during missions using this sensor package. The reagents are mixed with the seawater as they pass through a length of tubing before being pumped through a fluorometer explosives detector.



**Figure 2.2.4:** The RPUUV with Chemical Sensor Payload

## 2.2.4 Additional New Sensor Systems & Modifications (Tasks 3.4 & 3.5)

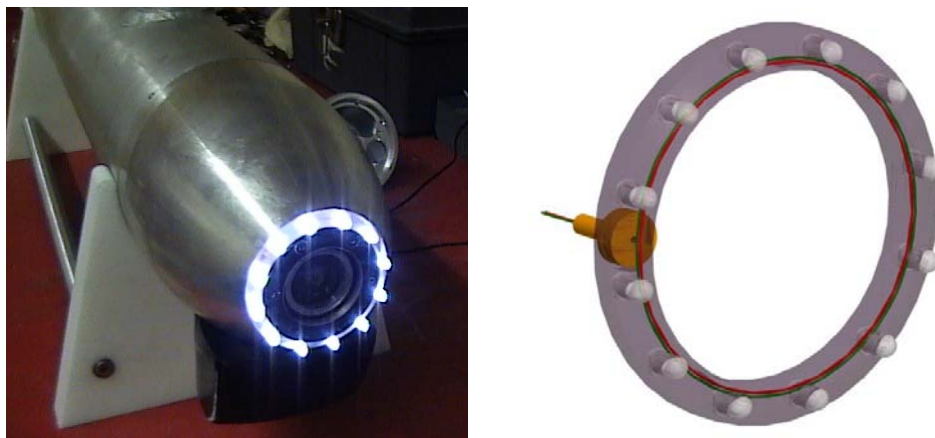
The original RPUUV that was developed in the first year of this work has had several modifications and additions over the course of year two. Modifications and improvements to the microcontroller software and operator interface GUI will be discussed in section 2.2.5. The major hardware changes are summarized in Figure 2.2.5 and described in this section.



**Figure 2.2.5:** Year 1 RPUUV Modifications

### 2.2.4.1 LED Lighting

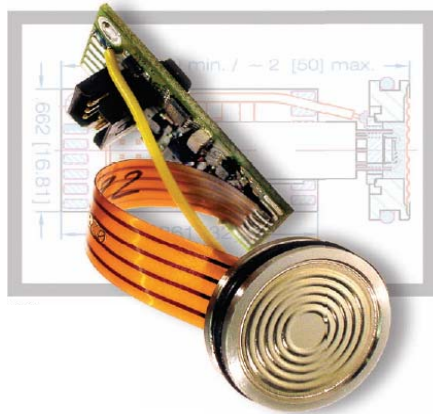
A ring of high brightness LEDs was designed, built and fitted to the nose of the RPUUV, surrounding the camera viewport as seen in Figure 2.2.6. These lights can be toggled on and off via a spare channel on the operators RC controller handset. The LEDs are arranged in a continuous ring but are actually wired as four sets of three so that combinations of illuminated groups and various flashing regimens can be used to signal selected vehicle states. (i.e. One group only comes on when the thruster is activated via the magnetic reed switch.)



**Figure 2.2.6:** LED Light Ring for the RPUUV

#### 2.2.4.2 Pressure Sensor

During sea trials of the RPUUV in year one, it was considered desirable to have a better indicator of the vehicles depth. An accurate depth sensor would also allow for automated depth following control algorithms to be incorporated in future revisions. To this end, an OEM pressure sensor was fitted to the RPUUV tail section. The selected sensor is a Series 30x OEM pressure transducer made by Keller America Inc. This small sensor, seen in Figure 2.2.7, has built in conditioning circuitry and a digital output format that can easily be read and controlled through a serial port on the vehicles embedded computer.



**Figure 2.2.7:** Keller America Series 30x OEM Pressure Transducer

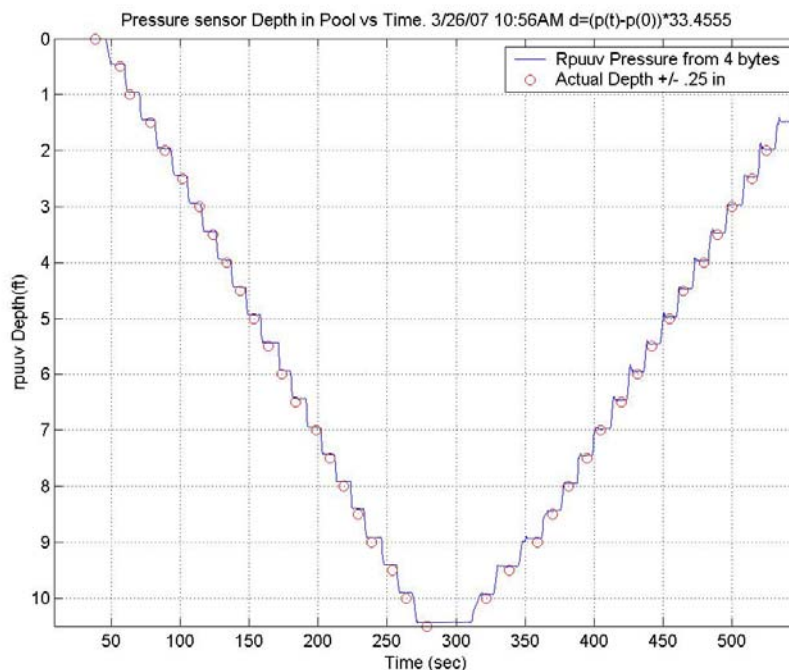
A pocket was machined in the tail end-cap assembly, between the two stepper motors, to house the transducer diaphragm, with a small hole penetrating the end-cap to allow the outside ambient pressure to impinge on the diaphragm. The mounting arrangement can be seen in Figure 2.2.8.



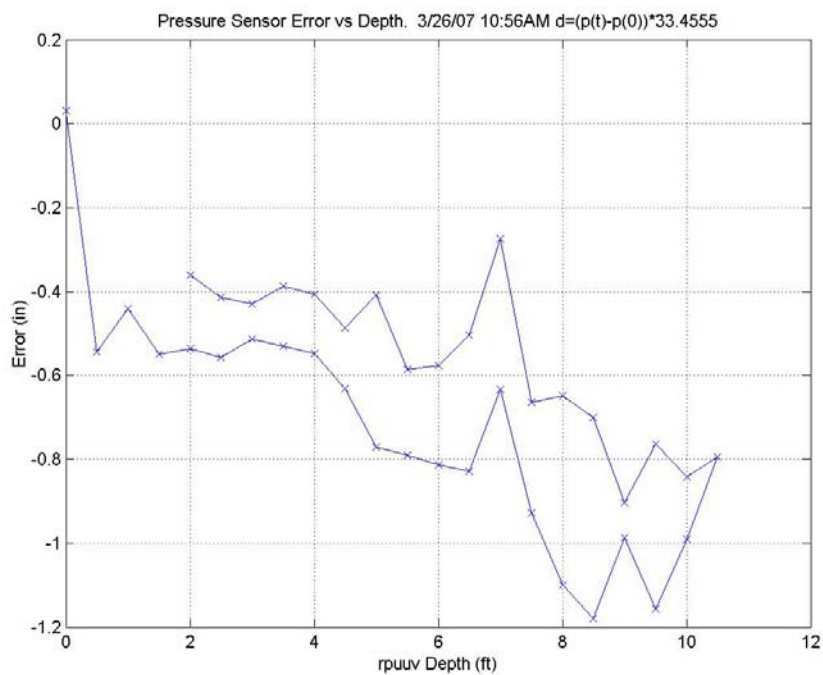
**Figure 2.2.8:** Pressure Sensor Integrated in RPUUV Tail Section



The installed pressure sensor was calibrated in a test tank at FAU and a comparison of measured to actual depth reading can be seen in Figure 2.2.9.



**Figure 2.2.9:** Pressure Sensor Calibration Results



**Figure 2.2.10:** Pressure Sensor Error vs. Depth

From these tests it was apparent that the error in the transducer reading increased slightly with depth but that at a depth of 10ft the error was less than 1.2 inches. The sensor error as a function of depth is presented in Figure 2.2.10.

#### 2.2.4.3 Altimeter

The original RPUUV developed in year one was designed to use the commercially available PC-View forward scanning sonar. After evaluation and testing of this sonar however, it was determined to be unsuitable for our application. Altitude information is critical however to successfully control any underwater vehicle, so alternatives were sought. The chosen replacement is an 8-channel obstacle avoidance sonar that is being developed by Dr. Steven Schock's group at FAU. The details of this sonar package were discussed in the CCST Year One Final Report. It essentially gives closest object information for each of 8 transducer beams, one of which is downward looking to give altitude, and its packaging is shown in section 2.2.4.4.



**Figure 2.2.11:** Cruz-Pro OEM Depth Sounder

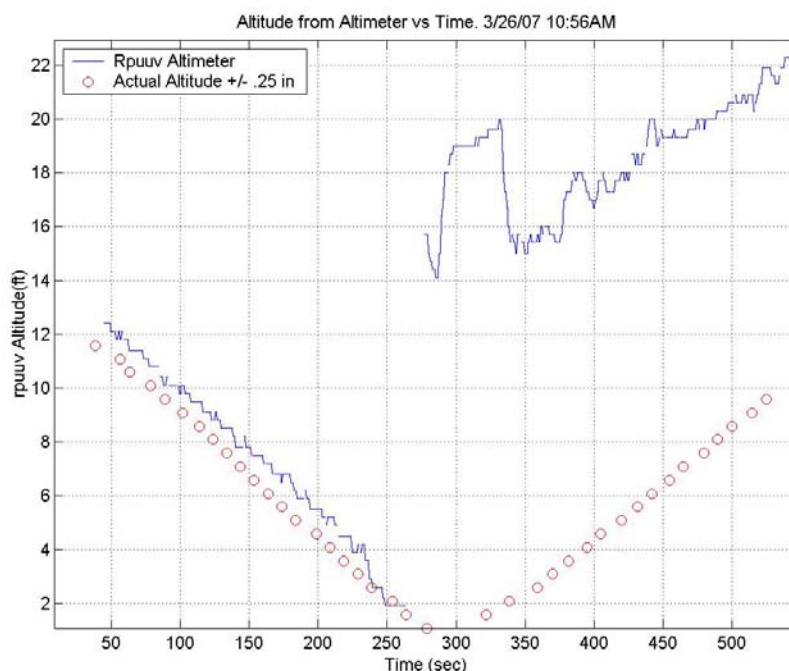
Since the long-term obstacle avoidance sonar solution is still under development, an intermediate sensor was installed to provide altitude measurements. A convenient solution that fits with minimal re-machining into the space vacated by the PC-View transducer is the Cruz-Pro Active Depth through-hull transducer. This altitude sensor provides a simple NMEA string output that can be read directly through a serial port on the RPUUV's embedded computer. An un-potted version was acquired from Cruz-Pro that was machined to fit the available space in the nose section of the RPUUV. Views of this transducer fitted to the nose section of the RPUUV can be seen in Figure 2.2.12.





**Figure 2.2.12:** Cruz-Pro Depth Sounder Installed in RPUUV Nose Section

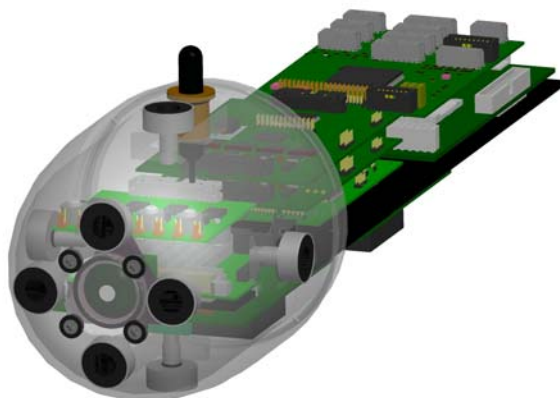
Subsequent testing of the RPUUV with the Cruz-Pro altimeter showed good results except when the vehicle gets too close to the bottom. At altitudes of less than about 2-3 feet, the sensor seems to get confused with surface reflections and the output altitude data jumps accordingly. This effect was duplicated in a test tank environment and the data can be seen in Figure 2.2.13. Solutions to this problem are currently being discussed with the manufacturer of this sensor, but resetting this device in the event of a sudden discontinuity in the bottom appears to be the only immediate solution.



**Figure 2.2.13:** Tank Test Calibration Results for Cruz Pro Depth Sounder

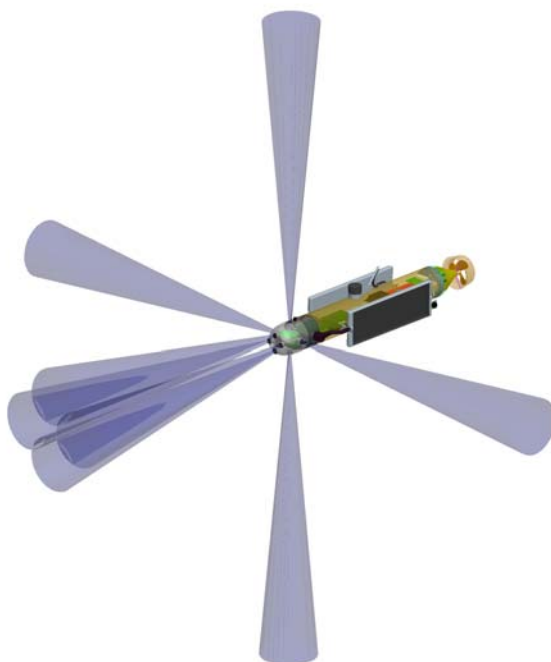
#### 2.2.4.4 Obstacle avoidance Sonar

As discussed in section 2.2.4.3, the PC-View scanning sonar from year one will be replaced with an 8 channel obstacle avoidance sonar package in year three of this work. However substantial progress has been made on this sonar system in year two, including the fabrication and acquisition of many of its components. Figure 2.2.14 shows a CAD model of the obstacle avoidance package with a new RPUUV nose-cone accommodating the 8 transducers. Packaging of the obstacle avoidance and imaging sonar electronics are also shown in this figure .



**Figure 2.2.14:** Obstacle Avoidance Sonar Packaging

The eight transducers have a beam width of about 10 degrees each (6dB down points) and are arranged with one looking upward, one downward, one looking to the left, one to the right, and four forward, as shown in Figure 2.2.15.



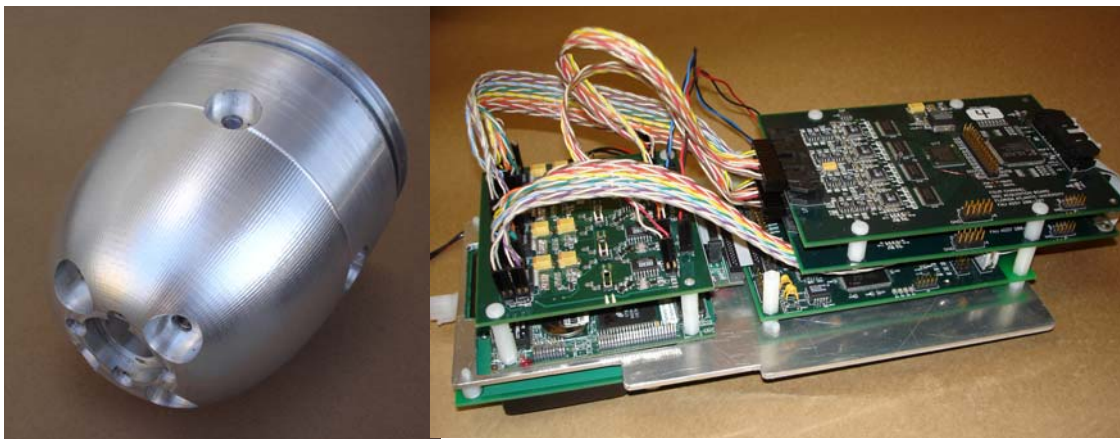
**Figure 2.2.15:** Obstacle Avoidance Sonar Beam Angles and Directions

The obstacle avoidance sonar transducers are all identical and interchangeable. Several of these transducers, seen in Figure 2.2.16, have been built and tested to determine their response and beam-width.



**Figure 2.2.16;** Obstacle Avoidance Sonar Transducers

A new nose cone has been fabricated to accommodate this sonar package and is shown in Figure 2.2.17 along with the prototype processing electronics boards.



**Figure 2.2.17:** Obstacle Avoidance Sonar Nose-Cone and Electronics

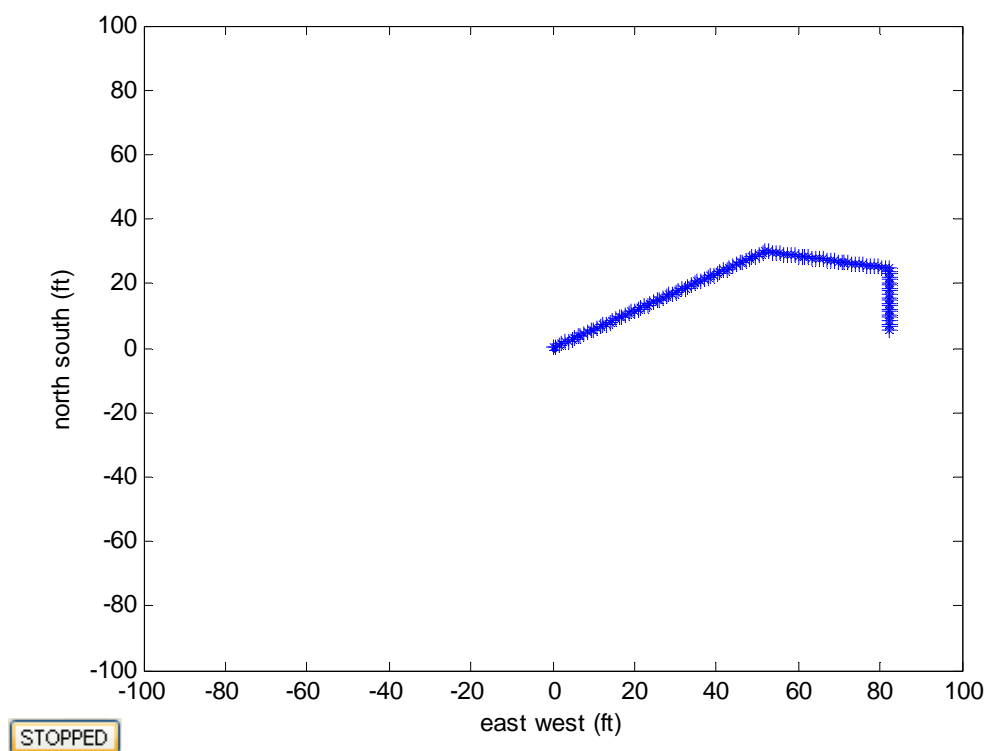
### **2.2.5 Topside Interface and Vehicle Simulations**

One of the most important features of the RPUUV is the topside interface with the pilot. At present the vehicle is controlled using a joy stick through an RF or acoustic link. Displaying information gathered by the vehicle and the vehicle status is only possible at this time using a WiFi link, and one of the objectives for year three of this project is to replace the WiFi link with a high speed acoustic link and positioning system. Some development work has been carried out in year two on the most suitable topside display for the system. This has been used in both vehicle simulation and during the experimental testing. Although more work needs to be done in this area, some interesting results have been obtained during year two of this project and these will be described in this section.

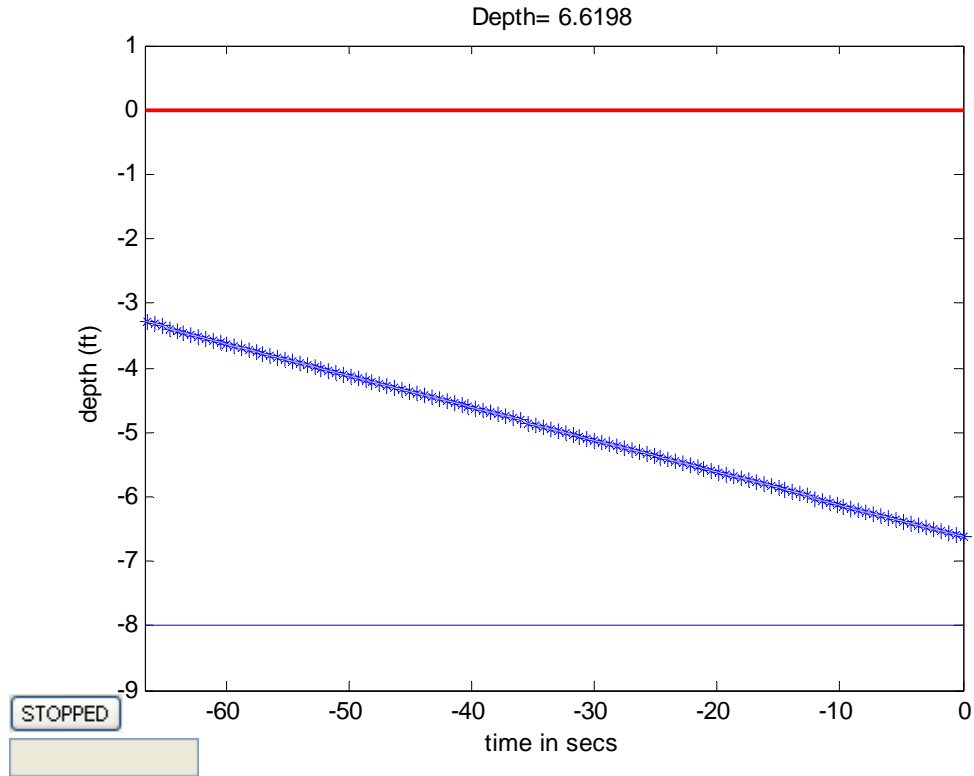
Each of the sensor systems built into the vehicle comes with its own software display package and the approach used at the end of year one was to utilize each of these kernels

running on the on board computer (the mother board). The topside user would then log on remotely through the WiFi link to view the multiple displays from the various sensors. In order to combine the outputs into a more ergonomic display the data needs to be combined into a single GUI. For development purposes the operating system chosen to combine the various signals was MATLAB. The reason for this choice was that the relative simplicity and universality of coding in MATLAB was seen as a distinct advantage. Once the GUI has been optimized, translating a MATLAB code into a lower level language is relatively straight forward. MATLAB was mounted on the mother board of both vehicles and codes were written to access data from each sensor through multiple serial and USB ports.

At the outset the vendor supplied GUI for the on board compass, pitch and roll sensor was displayed using a digital read out and an icon which gave a visual image of the vehicle attitude and heading. The first step in replacing this was to develop a MATLAB code that provided a compass rose and a vehicle attitude graphic. However it was found that this was difficult to follow during vehicle tests because it gave no indication of the rate of change of the vehicle heading and pitch. Following a series of simulation tests the display was changed to show a map of the vehicle position during the previous ten seconds. The display consists of two maps as shown in Figure 2.2.18 and 2.2.19. The first figure (Figure 2.2.18) gives the relative Lat-Long position of the vehicle, while the second figure (Figure 2.2.19) gives the depth of the vehicle based on the time integrated pitch of the vehicle. Both these results were calculated using an assumed forward speed.



**Figure 2.2.18:** The Topside GUI Display Showing the Vehicle Track



**Figure 2.2.19:** The topside GUI display showing the vehicle depth and bottom location

This was effective for the compass reading providing a visual measure of the vehicle position was available. However the depth estimate was a concern and so the output of the altitude sensor was also placed on the same display. The altitude sensor proved to be unreliable, as is described in section 2.2.4.3, and so it was decided that the depth display should also include the output of a pressure sensor which would give the depth of the vehicle. This sensor was installed in the vehicle and its output is displayed on the GUI as shown in Figure 2.2.19.

The advantage of the existing system is that it is now relatively straight forward to build in automated control of the vehicle based on the sensor data, but this will be considered during the vehicle development tasks which take place during year three of the project.

Using assumed vehicle data a simple vehicle simulator was also developed during year two and this also aided the development of the GUI. The task of the simulator was to navigate the vehicle towards a seawall, make a right turn so the vehicle proceeded parallel to the wall and then to make another right turn. This simple simulator indicated that the vehicle controls needed to be updated at least twice a second, and that the vehicle needed to be operated at a slow speed of less than 1 knot. It was also clear that joystick controls would be superior to on screen buttons to control the vehicle direction.



### 2.2.6 In Water Testing (Task 3.5)

Figure 2.2.20 below, shows the equipment that is required to run in-water operations with the year 1 RPUUV. Currently the only item that requires external power is the topside WiFi Ethernet Bridge. A small DC battery supply will be built to power this unit in year 3, enabling operations with this system to be conducted in any remote location where a 110V supply may not be available.

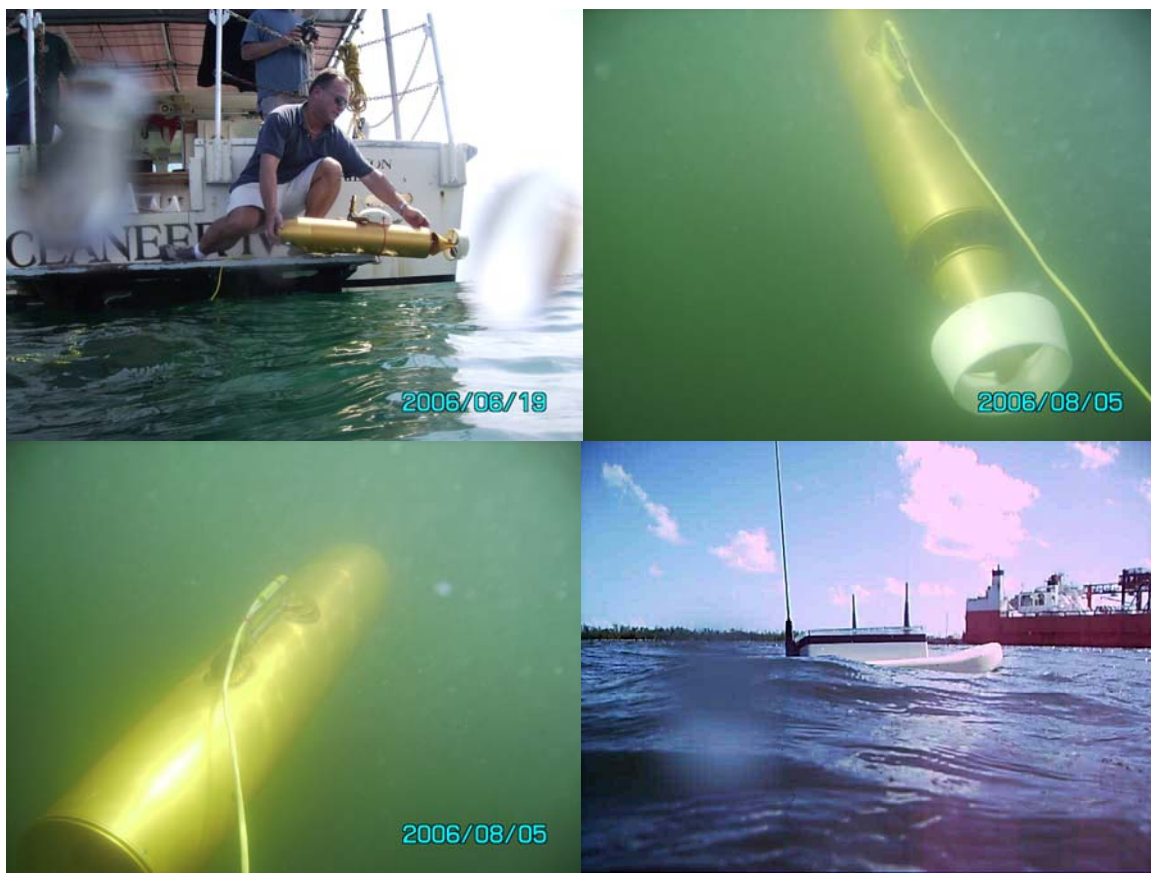


**Figure 2.2.20:** Operation Equipment for RPUUV Testing

To date, testing with the RPUUV has been conducted from the back of a small research vessel, both dockside in a marina and in a Port Everglades turning basin, and from the shore of a small man-made lake on the FAU Boca Raton campus. Year 2 testing is summarized below:

#### ***Summer 2006 Port Everglades Testing***

Tests were conducted in about 40ft of water in a turning basin of Port Everglades, Florida. In these trials the RPUUV was equipped with the Cruz-Pro depth sounder. While the information from this sensor proved very useful in helping the operator control the vehicle depth, it proved to be intermittent and jumpy depending on the bottom type and closeness of the RPUUV to the bottom. It was thus decided that the addition of a pressure sensor would make depth control more accurate and reliable. Additionally it was concluded that the forward looking camera could not be used as a navigational aid in turbid water environments and that the top-side presentation of the data seen by the operator needed refinement by condensing these data into a single GUI display to reduce the operators feeling of information overload.



**Figure 2.2.21:** Photographs from Port Everglades Trials

### *November 2006 Trials*

More testing was performed in the port environment to evaluate different GUI formats for the top-side data display. It is concluded that once the operator loses visual contact with the vehicle it is extremely difficult to confidently navigate the RPUUV through turbid water in a confined environment just using the topside console information. Automated depth and/or heading control by the vehicle's onboard computer is considered to be a desirable asset to the operator.



**Figure 2.2.22:** Photographs from November 2006 Port everglades testing

### ***December 2006 Lake Tests with Surface piercing Mast***

To try to give the RPUUV operator a better visual reference of the vehicle's depth and heading, an 8ft long graphite mast was added to the top of the RPUUV. This mast was graduated with taped markers every 2ft. Although this approach limited the depth at which the vehicle could operate, it was found to be a valuable visual cue allowing the operator to more easily control the vehicle depth in shallow water environments and hence increased confidence to navigate underwater in a confined area.



**Figure 2.2.23:** Lake Testing with Surface Piercing Mast

Significant trim adjustments were necessary however to maintain level flight because of the varying drag of this appendage, although at a slow constant speed it was possible to maintain constant depth and navigate the RPUUV quite accurately.

### ***April 2007 Testing with Modem Control***

In April 2007 the first tests were conducted with the year 2 RPUUV under modem control. In these tests the tow-float RF communications channel between operator and vehicle was replaced with a purely acoustic link via an acoustic modem. The tested vehicle had no other sensors mounted to it so operations were conducted with the vehicle suspended about 2ft below a small foam float. A slightly negatively buoyant vehicle ensured that the modem transducer mounted on the top of the pressure hull would remain below the water surface at all times.

The RPUUV was launched and recovered from the back of a small research vessel in the SeaTech marina as seen in Figure 2.2.24. In this configuration the vehicle proved to be very controllable and the acoustic link sufficiently robust to drive the submersible out to a range of about 75m without losing contact. The vehicle was programmed to simply stop the thruster and return the vectored tail to zero deflection should the acoustic link be lost for more than a few seconds. Some latency between operator commands and vehicle response was apparent, but this was manageable given the constant visual link between



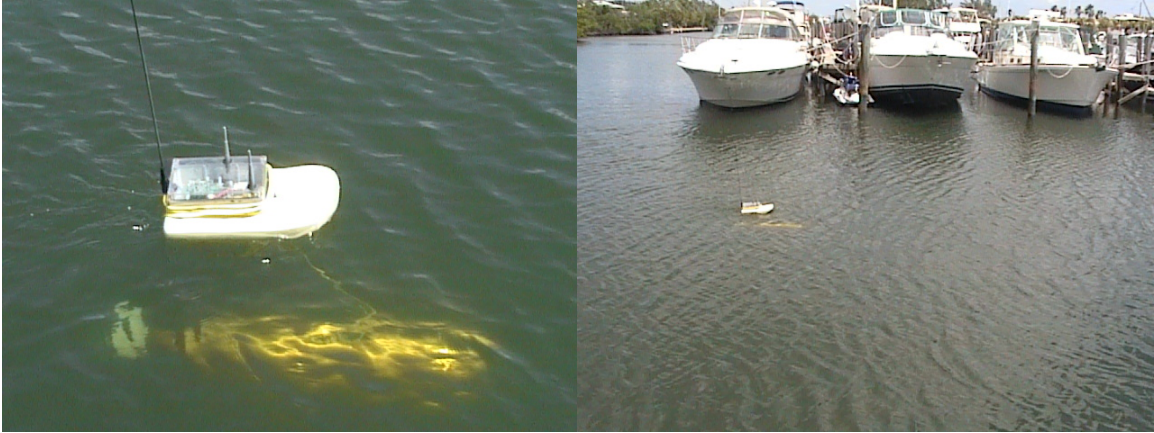
operator and vehicle. However, in this configuration it was possible to accurately pilot the vehicle underwater alongside and close to a stationary boat or seawall.



**Figure 2.2.24:** Testing of Year 2 RPUUV with Modem Control Only

#### *April 2007 testing with short tow-float cable*

After the successful modem tests with the year 2 RPUUV suspended on a short tether below a small float, it was decided to try a similar approach with the regular RF tow-float controlling the vehicle depth. Again the vehicle was ballasted to be slightly negatively buoyant, and horizontally trimmed so that it would hang about 3 ft below the tow-float as seen in Figure 2.2.25. In this configuration the vehicle proved to be extremely controllable. With the visual directional indicator of the tow-float and automatic depth control provided by the float, the operator's task was reduced to one of simply direction and speed control. Some trimming of the vehicle was necessary to overcome the drag forces of the tow-float cable at higher speeds, but at low speed the operator was able to maneuver the vehicle very accurately and confidently. Further investigation of this approach with longer tether lengths will be investigated in year 3 of this research.



**Figure 2.2.25:** Marina Testing of RPUUV with Short Tow-Float Cable

## 2.3 Acoustic Communications

**PI: Dr. P. Beaujean**

### Task 3.7

#### 2.3.1 Summary

The main objective of this portion of the project is to develop communication systems for the purpose of transmitting and receiving information wirelessly between a user and the Remotely Piloted Underwater Vehicle (RPUV). Transmitted information is used to pilot the RPUV and relay its position. Information received from the RPUV combine acoustic images of the environment and status report of the vehicle. During the first year of this project radio wave (WiFi) communication was used to control the vehicle. Whenever the tow-float solution becomes impractical, a slower but fully wireless acoustic modem is to be used. The design must consider the issues associated with acoustic communications in port at high data rates, using a high-frequency acoustic modem, and the piloting and tracking of the RPUV, using a command-and-control acoustic modem.

Dr. Beaujean is responsible for this acoustic communication project. Two graduate students have been supported to assist with the development and analysis of the communication system. The deliverable consisted in a report presenting the design and testing of an acoustic wireless communication system to control and retrieve data from the RPUV.

#### 2.3.2 Introduction

##### **Operational criteria:**

The objective of this research is to be capable of piloting an underwater vehicle remotely using acoustic communications. This vehicle is to perform search missions, principally in ports and very shallow waters, to find potentially dangerous or illegal objects such as explosive or narcotics. Note that this vehicle can also perform scientific missions. In its initial configuration, the vehicle is equipped with:

- An imaging sonar system, a camera and an optional chemical sensor for threat detection.
- A tilt sensor and compass, and an Ultra-Short Baseline acoustic positioning system.
- An embedded processor, a motherboard and Ni-Mh batteries.
- A tow-float with a WiFi access point (802-11g, 2.4 GHz) for image and data transmission and Radio Frequency (72 MHz) control unit for piloting.

In a second, fully untethered configuration, the vehicle is equipped with the acoustic communication package:

- A low-speed acoustic modem for remote piloting and positioning (surface to vehicle).
- A high-speed acoustic modem for image transmission (vehicle to surface).

### **Acoustic remote piloting and positioning:**

The criteria retained for the piloting and positioning of the RPUV are as follows:

- The vehicle is to operate for approximately 2 hours in approximately 1 to 20 m of water, in the proximity of walls, pilings and underneath ships.
- The vehicle is moving at a top-speed of 1 m/s, at a maximum range of 100 m.
- The vehicle is assumed to remain at least 0.25 m from the surface during operations.
- The peak power consumption of the modem receiver unit in the vehicle is to be kept to a minimum (0.5 W) and use as few transducers as possible (one ITC-3460 and one ITC-1089D).
- At the surface, the source level is limited to 168 dB re 1 $\mu$ Pa/1m averaged over time, due to environmental requirements, and must not interfere with the imaging sonar nor the USBL.
- The remote piloting modem must operate so that it can easily replace the remote piloting unit initially used for piloting.
- The pitch, yaw and thrust of the vector thruster unit must be updated at least 3 times every two seconds, with 128 positions for pitch and yaw, and 128 levels of thrust.

### **High-speed acoustic communications:**

The criteria retained for high-speed acoustic communication system, used to transfer video and sonar information, are as follow:

- A maximum achievable data rate of 87,768 bits per second at fairly close range (150 meters and less) in harbors and in very shallow water.
- Small, low-power and inexpensive device, well suited for modern untethered underwater vehicles operating in very shallow water and ports.
- Compressed video or high-resolution sonar images should be relayed to a topside unit in real-time.

## **2.3.3 Acoustic remote piloting and positioning:**

### **2.3.3.1 System overview:**

The objective is to pilot the vehicle using sound. In this configuration, the RPUV does not use a tow-float. Instead, two acoustic communication units are used for piloting and navigation, and to relay video and images. An overview of the acoustically-piloted RPUV is shown in Figures 2.3. 1 and 2. Figure 2.3.3 shows the various components of the RPUV used for communications and piloting.

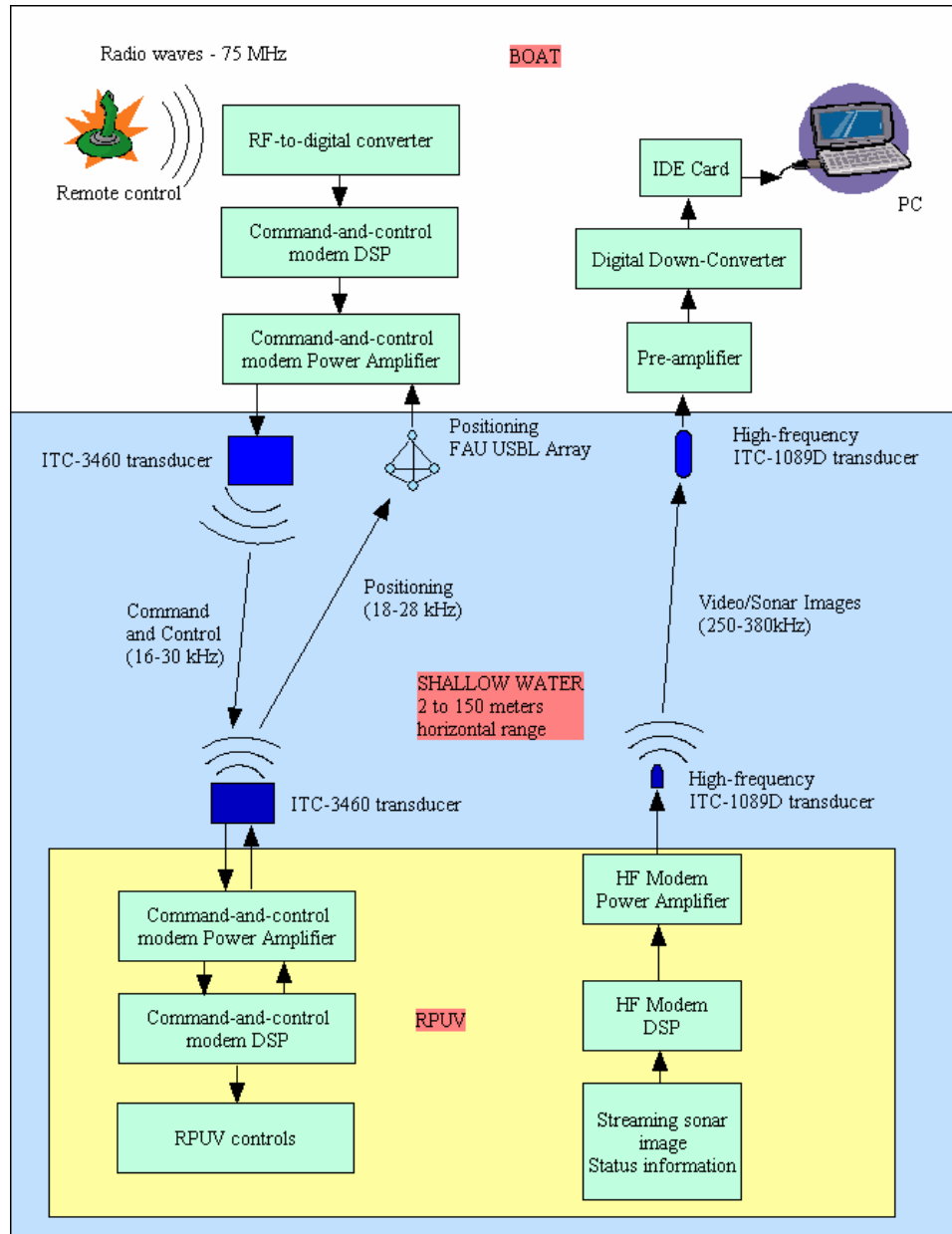


Figure 2.3.1. Overview of the RPUV control using a tow-float and acoustic waves.

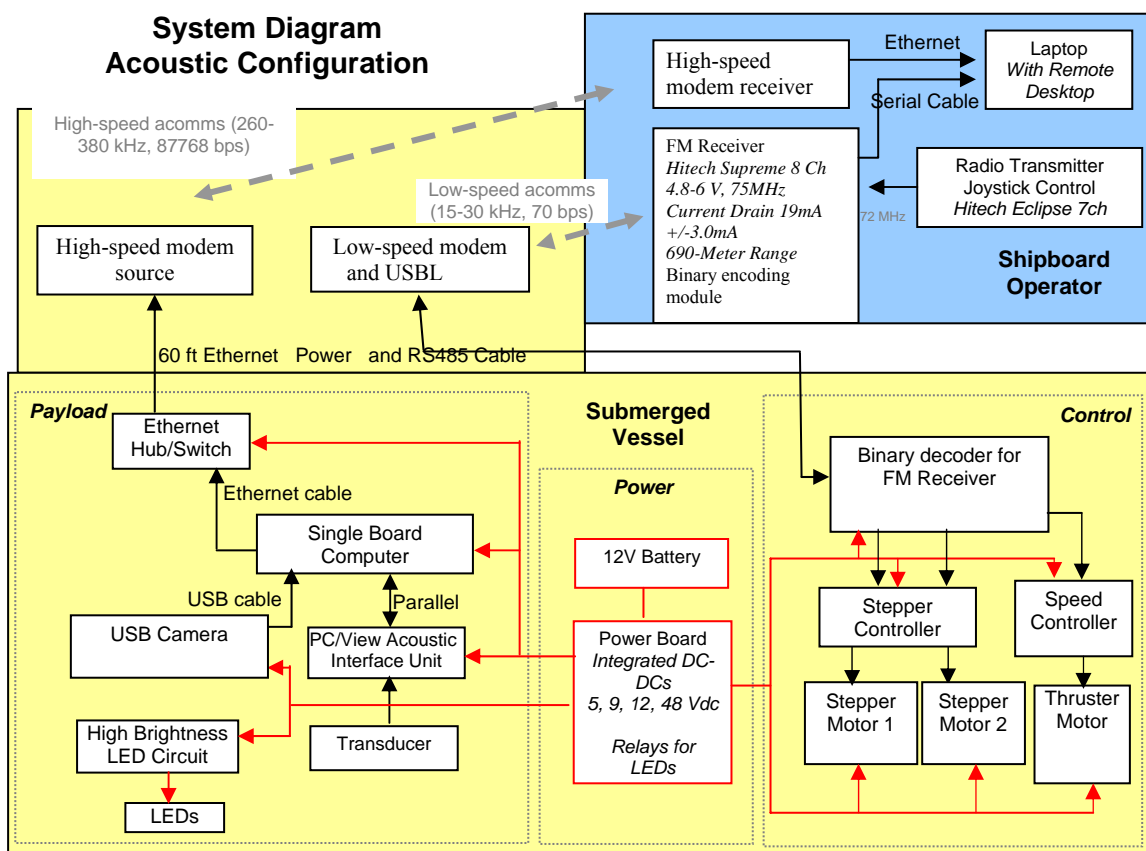


Figure 2.3.2. Detailed diagram of the RPUV control using acoustic waves.



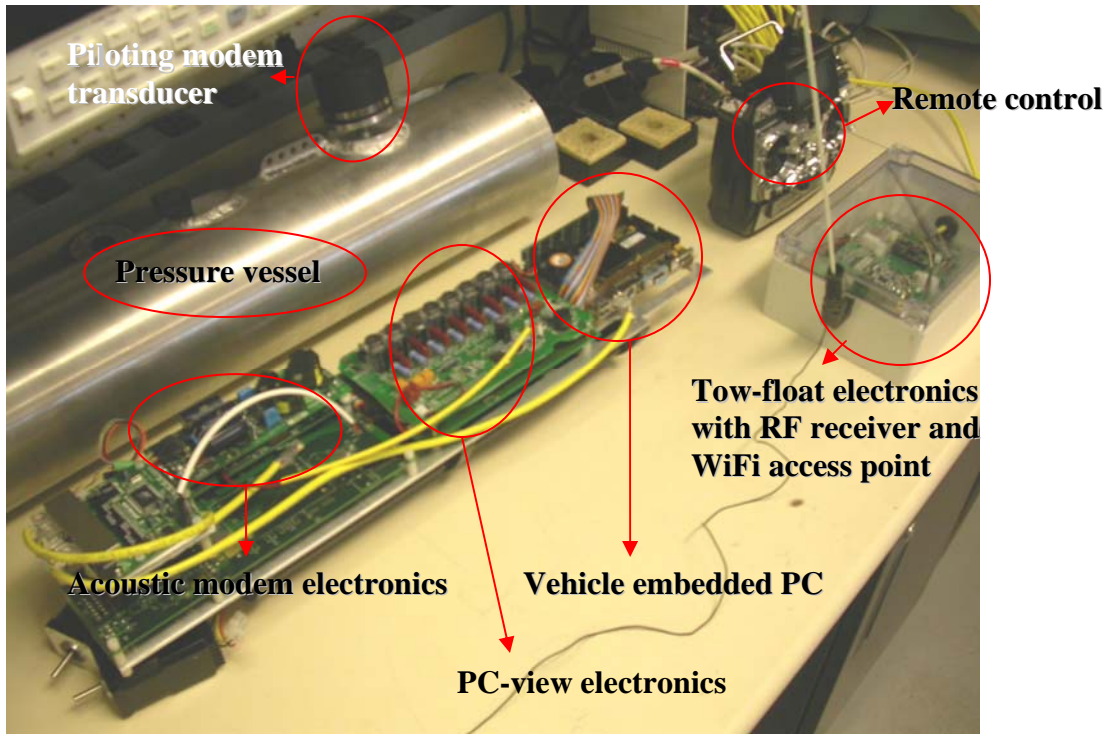


Figure 2.3.3. The remote control components of the RPUV.

Using acoustic communications to send command and control information to a UUV is not a unique concept [2], though previous attempts at remote piloting have not taken into consideration the concept of real-time command and control. The technique used most often is supervisory control instead of true joystick-type remote control [3]. This sort of control is used mostly to send new autopilot algorithms or preprogrammed sequences from the pilot to the UUV [4]. One example of true joystick-type remote control is given in [5], where the vehicle is piloted once on the surface of the water using a wireless RS-232 uplink, to aid in the launch and recovery of the vehicle and is only functional on the surface. The concept of full joystick-type command and control of a UUV using acoustic communications is a novel concept that is being developed in this program.

The piloting and positioning unit uses an FAU-Dual Purpose Acoustic Modem (DPAM) [6][7], which transmits remote-piloting commands from the topside to the underwater vehicle and receives position information back from the vehicle periodically. The topside unit is equipped with an ITC-3460 reciprocal transducer for piloting. Acoustic positioning takes place using a top-side FAU Ultra-Short Baseline (USBL) array [8][9]. The high-level communication and navigation algorithms for both topside and vehicle modem units are shown in Figures 2.3. 4 and 5. A picture of the FAU DPAM electronics is shown in Figure 2.3.6. The FAU USBL array and IMU are shown in Figure 2.3.7.

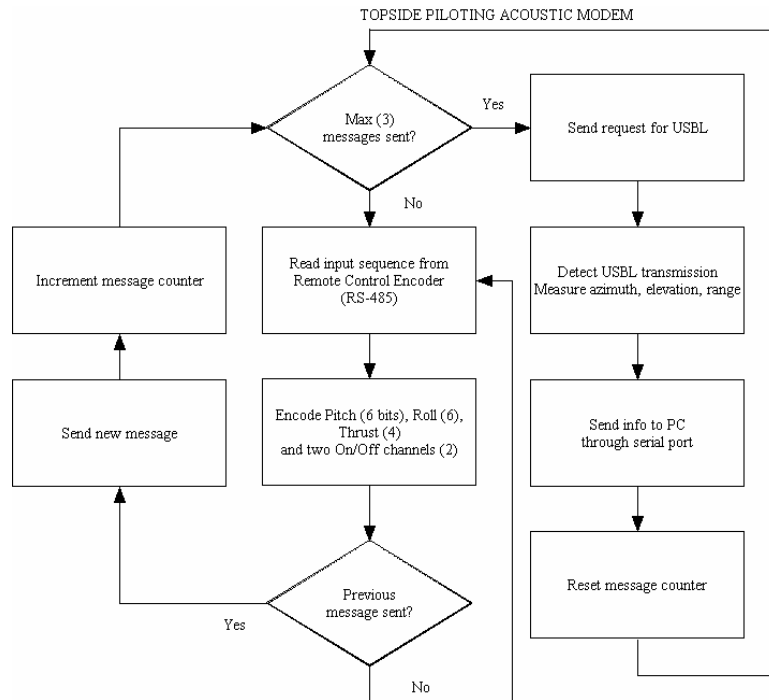


Figure 2.3.4. High-level flow chart of the acoustic piloting and positioning at the user end.

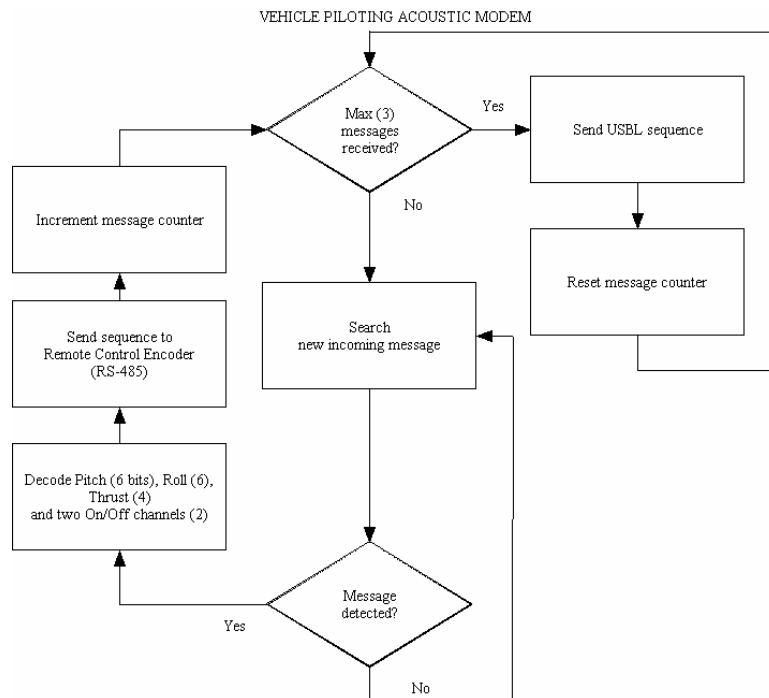


Figure 2.3.5. High-level flow chart of the acoustic piloting and positioning at the RPUV end.



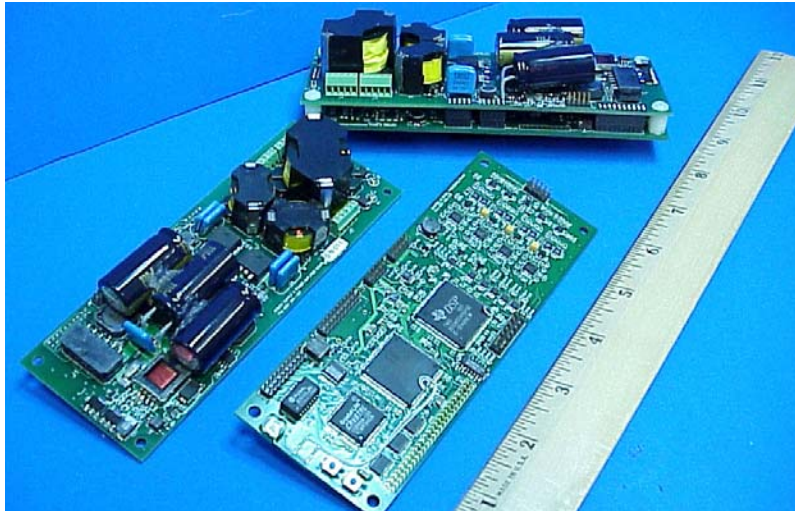


Figure 2.3.6. Acoustic remote piloting electronics.



Figure 2.3.7. USBL positioning array (left), coupled IMU and USBL Array (center) and XSens MTi IMU (right).

At present, the remote piloting unit is capable of transmitting up to 3 piloting messages every two seconds, while the vehicle position is updated approximately once every 2 seconds. The tested range for piloting is approximately 30 m, with an estimated maximum range of 3000 meters at full power based on previous experimentation of the FAU DPAM. The electronic units are built, and the RPUV is already equipped with an FAU DPAM electronic DSP card and amplifier, and an ITC-3460 transducer. The FAU USBL unit has also been assembled. The first revision of the remote piloting software source and receiver has been completed and tested. The real-time positioning software is under test.

### 2.3.3.2 Remote acoustic piloting processing and experimentation:

Figure 2.3.8 provides the details of the remote piloting signal processing and hardware platform.

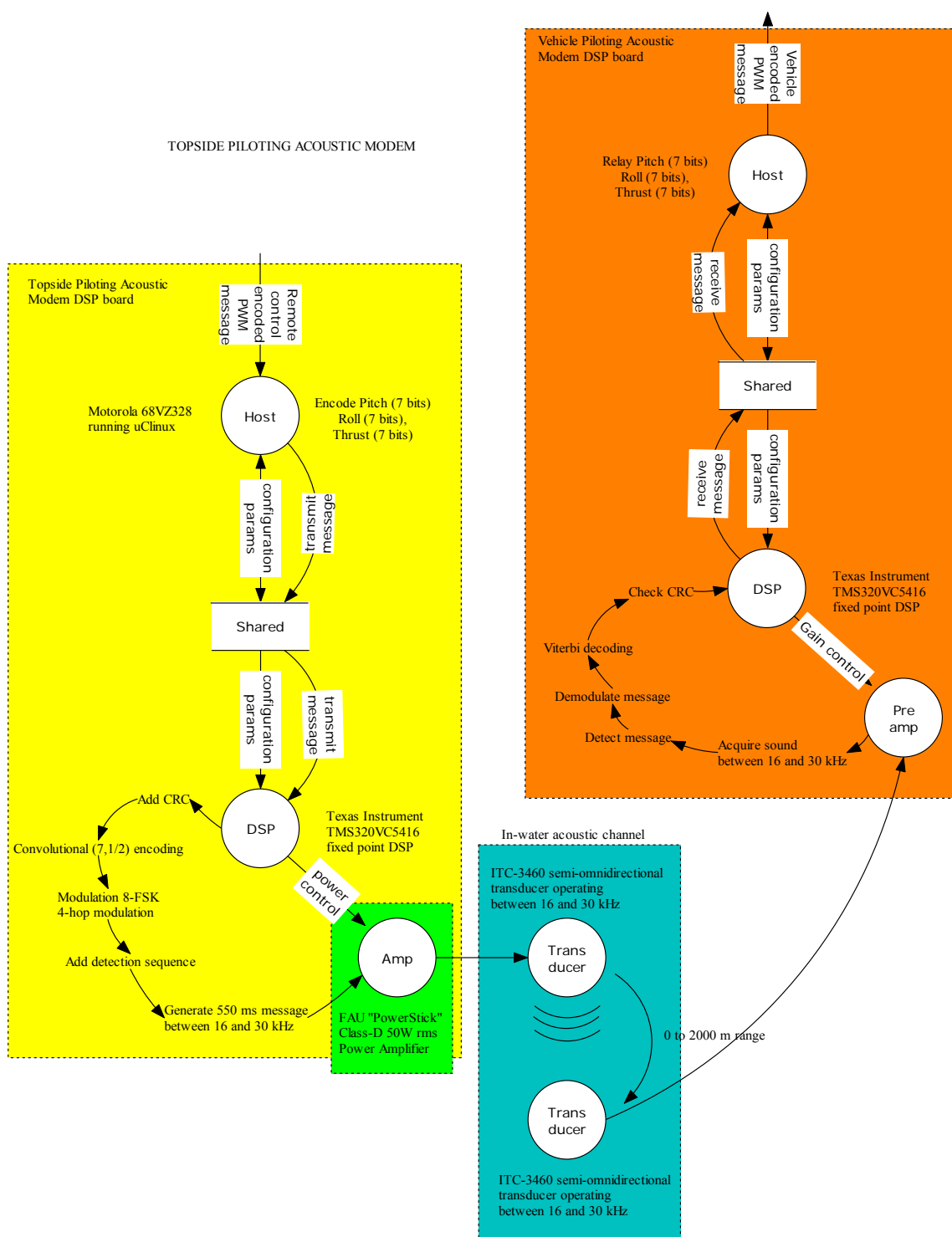


Figure 2.3.8. Detailed system diagram of the acoustic piloting software and hardware.

The modulation technique used for acoustic remote piloting is the robust M-ary Frequency Shift Keying (M-FSK) [6][7]. For this, the transmitted symbol  $j$  is defined as:

$$\tilde{m}_j(t) = 10^{SL/10} e^{i2\pi\left(f_{FSK,j} + \frac{W_s t}{2T_s}\right)t} \mu Pa \quad (1)$$

where  $f_{FSK,j}$  is the lower frequency of the  $M$ -possible frequency bands,  $SL$  is the source level,  $W_s$  is the symbol bandwidth, and  $T_s$  is the symbol duration.

The technique that is applied for demodulating the received signal is a cross-correlation function (or matched-filtering). This uses the non-coherent (energy) estimation to demodulate each symbol. The demodulation of each detection symbol is performed by cross-correlating the incoming symbol and a set of complex reference symbols. The cross-correlation function is computed as:

$$R_{\tilde{r}_j, \tilde{s}_p}(\tau) = \frac{1}{2T_s} \int_{-T_s}^{T_s} \tilde{r}_j(t) \tilde{s}_p(t + \tau) dt \quad j=1,2; p=1,\dots,32 \quad (2)$$

where  $\tilde{r}_j(t)$  is the incoming symbol occupying one frequency band centered on frequency  $f_j$ , a set of known complex reference symbols  $\tilde{s}_p(t + \tau)$ , and the symbol duration  $T_s$ . The peak magnitude of each cross-correlation is retained

$$c_j = \max \left\{ \left| R_{\tilde{r}_j, \tilde{s}_p}(\tau) \right| \right\}_{-T_s \leq \tau \leq T_s} \quad (3)$$

where  $c_j$  is defined as the symbol number when the peak correlation occurs. The location of the symbol associated with the largest peak magnitude is used for synchronization. In order to reduce the impact of fading due to acoustic multipath, frequency hopping is used. With this technique, successive symbols occupy different frequency bands, reducing the risk of inter symbol interference and frequency selective fading [6][7]. The modulated symbols are multiplied by a complex exponential that changes the frequency with time. The transmitted symbols then become:

$$\tilde{s}_j(t) = \tilde{m}_j(t) e^{i2\pi f_{HOP,j} t} \quad (4)$$

where  $f_{HOP,j}$  is the hopping frequency for a given hopping mode and symbol number. In the present case of acoustic remote piloting, the number of hops is 4 symbols, and 3 bits are contained in each symbol.

The acoustic piloting unit was first tested on a work bench using direct wired connection between the topside and the vehicle. Following this first series of successful tests, the acoustic piloting system was tested with transducers placed in a bucket. Finally, the unit was successfully tested in a small pool (Figure 2.3.9).



Figure 2.3.9. Remote acoustic piloting pool test.

Following this series of preliminary tests, the RPUV was tested in the marina at the FAU SeaTech campus in Fort Lauderdale, Florida. The vehicle was deployed from the stern of the R/V Oceaneer IV (Figure 2.2.24 in section 2.2), towing only a tow float made of syntactic foam. Additional weight was added to make sure that the vehicle remained fully submerged (slightly negative buoyancy) in order to keep the acoustic piloting transducer below the sea surface.

The RPUV was deployed and then recovered to power on the thruster motor. Once fully powered, the vehicle was on its way (Figure 2.3.10). The pilot maneuvered the RPUV around the marina in figure-8 patterns. The pilot then maneuvered the vehicle out to the canal (Figure 2.2.24). The vehicle was piloted in this area for eight minutes up to 70 meters away. After deeming the test successful, the pilot returned the vehicle to the Oceaneer IV and recovered the RPUV (Figure 2.2.24). An aerial view of the experiment is given in Figure 2.3.11.



Figure 2.3.10. Acoustic piloting in SeaTech marina, pilot view.



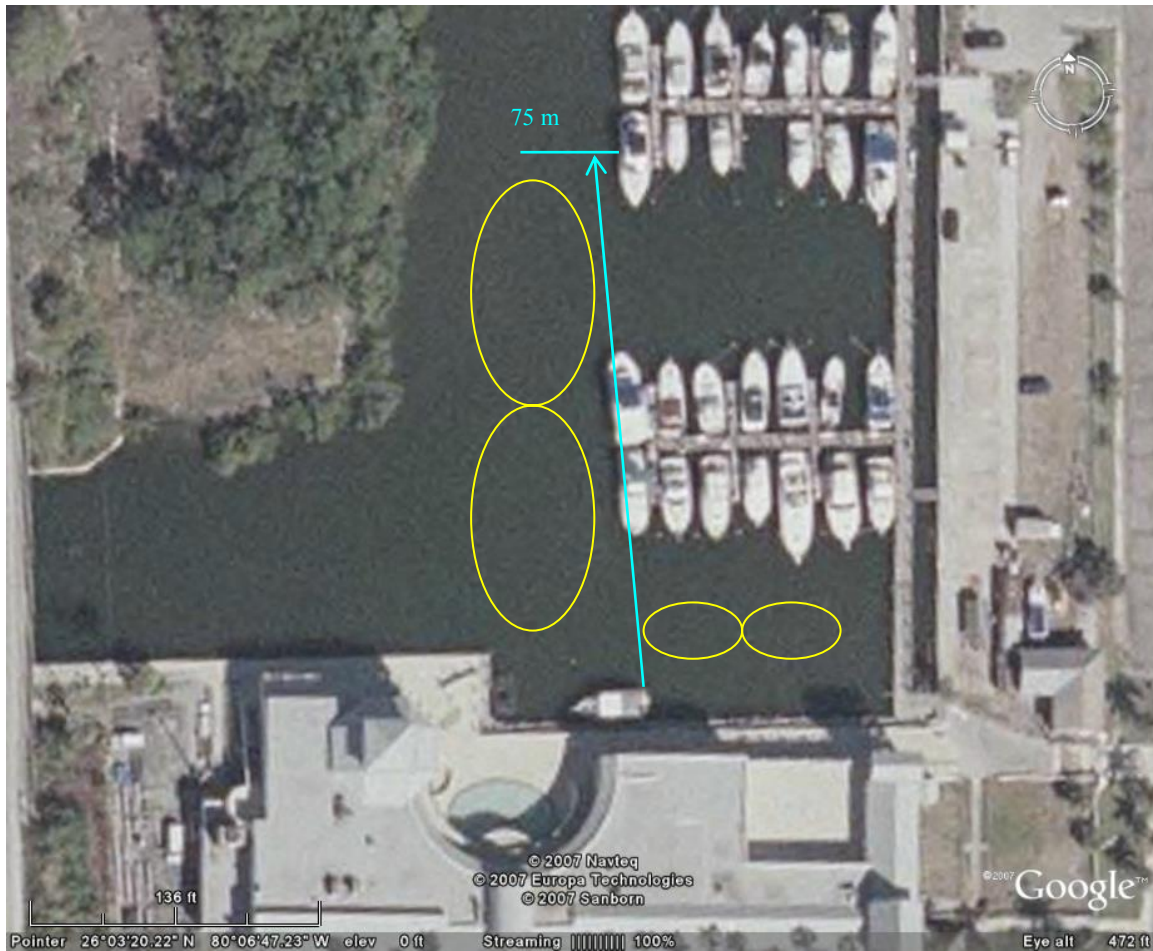


Figure 2.3.11. Aerial view of the FAU SeaTech marina.

### 2.3.3.3 USBL processing and experimentation:

The Ultra-Short Baseline (USBL) Acoustic Positioning System (APS) is mounted on a surface vessel, which implies that the position of the tracked vehicle is expressed in the frame of the ship, or navigational frame. But this position is actually not exploitable as the ship cannot keep a fixed position at the surface of the sea. The boat moves following the motion of the waves, the wind and also the current, or more simply because the crew wants to move the boat. It is necessary to transform the USBL APS measurements in the north east down frame (NED frame, Figure 2.3.12) using an XSens MTi IMU, capable of sensing the motion of the platform.

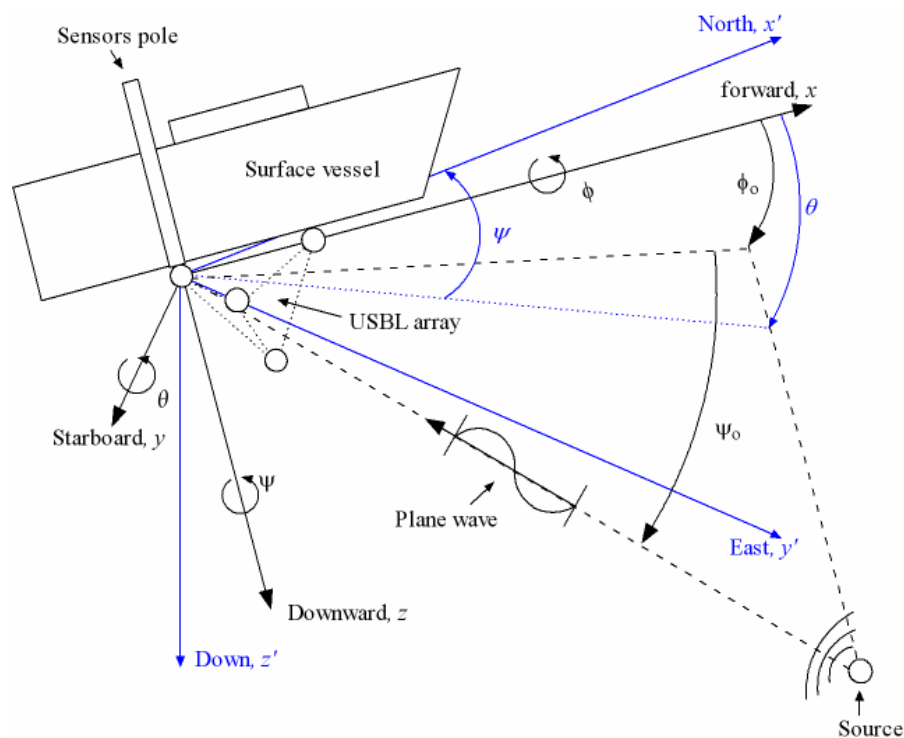


Figure 2.3.12. Frame transformation and motion compensation of the USBL system.

A Kalman filter is applied to this discrete linear dynamical system (Figure 2.3.13). The filter is of the form  $X(k) = F X(k-1) + w(k)$ , where  $X(k)$  is the state vector at time  $k$ ,  $F$  is linear operator defining the transition between to consecutives states and  $w(k)$  is the state noise [10].  $F$  defines the relation between the state vector  $X(k)$  at time  $k$  and the state vector at time  $k-1$ . The observation of the state vector is expressed as  $Z(k) = HX(k) + v(k)$  where  $Z(k)$  defines the measurement vector,  $H$  defines the measurement matrix and  $v(k)$  defines the measurement noise. The Kalman filter estimates the new state from the previous state and the current measurement while following the two phases defining the prediction and update Kalman algorithm. During the prediction phase, the estimation at the previous time step is applied to the state equation to predict the actual state, followed with the update phase where the new estimated state applied to the measurement equation to provide an updated state of the system. The state of the filter is defined as following:

- $\hat{X}^-$  or  $\hat{X}^+$  where  $\hat{X}^-$  is the predicted state estimate and  $\hat{X}^+$  is the updated state estimate
  - $P^-$  or  $P^+$  where  $P^-$  is the predicted error covariance and  $P^+$  is the updated error covariance. This error covariance matrix defines the accuracy of the state estimate.
- The Kalman has two distinct phases; the prediction and the update. At the prediction stage the filter computes a predicted state estimate  $\hat{X}^-$  using the previous estimation and the state equation. It also predicts error covariance  $P^-$  using the previous error covariance estimation and the state noise covariance  $Q(k)$ . The filter then updates the predicted estimates using the measurement weighted by a coefficient called Kalman gain. The

Kalman gain  $K$  is defined in such a way that the elements along the diagonal of the error covariance matrix are minimal, since these terms represent the variances for the elements of the state vector [10]. The updated state estimate  $\hat{X}^+$  depends on the Kalman gain and the error between the prediction and the measurement. The filter also updates the state covariance  $P^+$  using the state covariance prediction and the Kalman gain.

The sensors are mounted on a mast (Figure 2.3.14) currently installed on a small kayak for testing purposes (Figure 2.3.15). Later on, the same mast is to be installed on the actual test platform, expected to be the FAU Research Vessel Oceaneer IV.

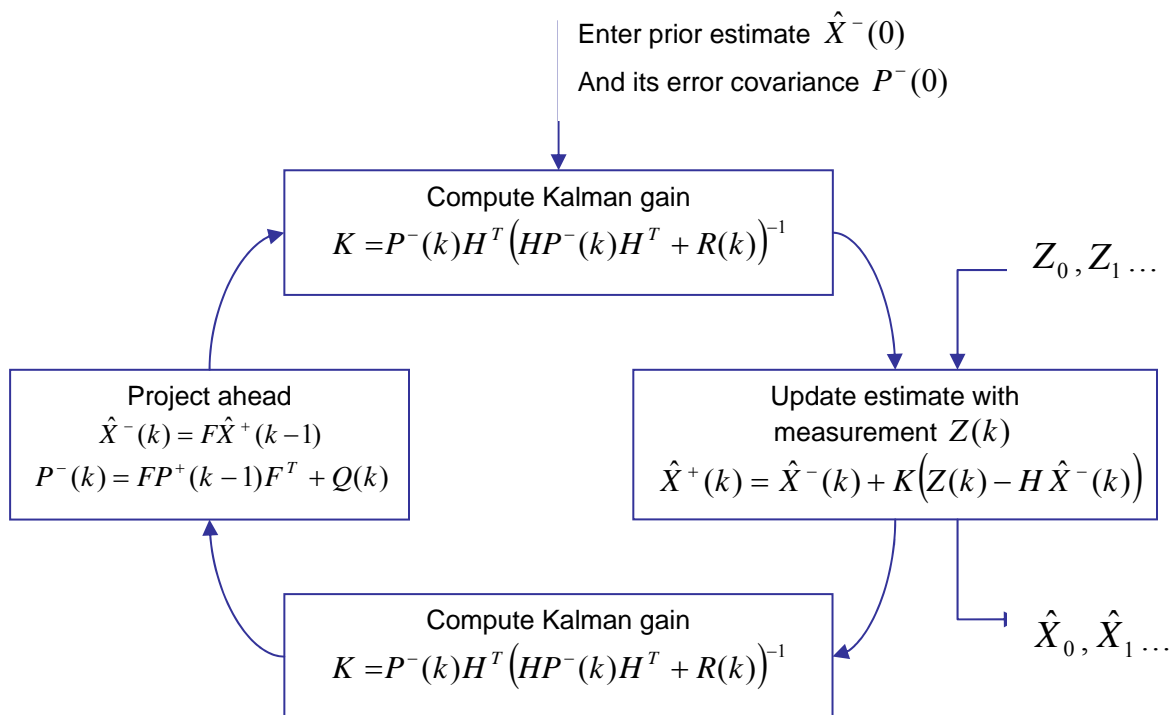


Figure 2.3.13. Kalman filter for USBL motion compensation.



Figure 2.3.14. USBL Mast mounted on a kayak.



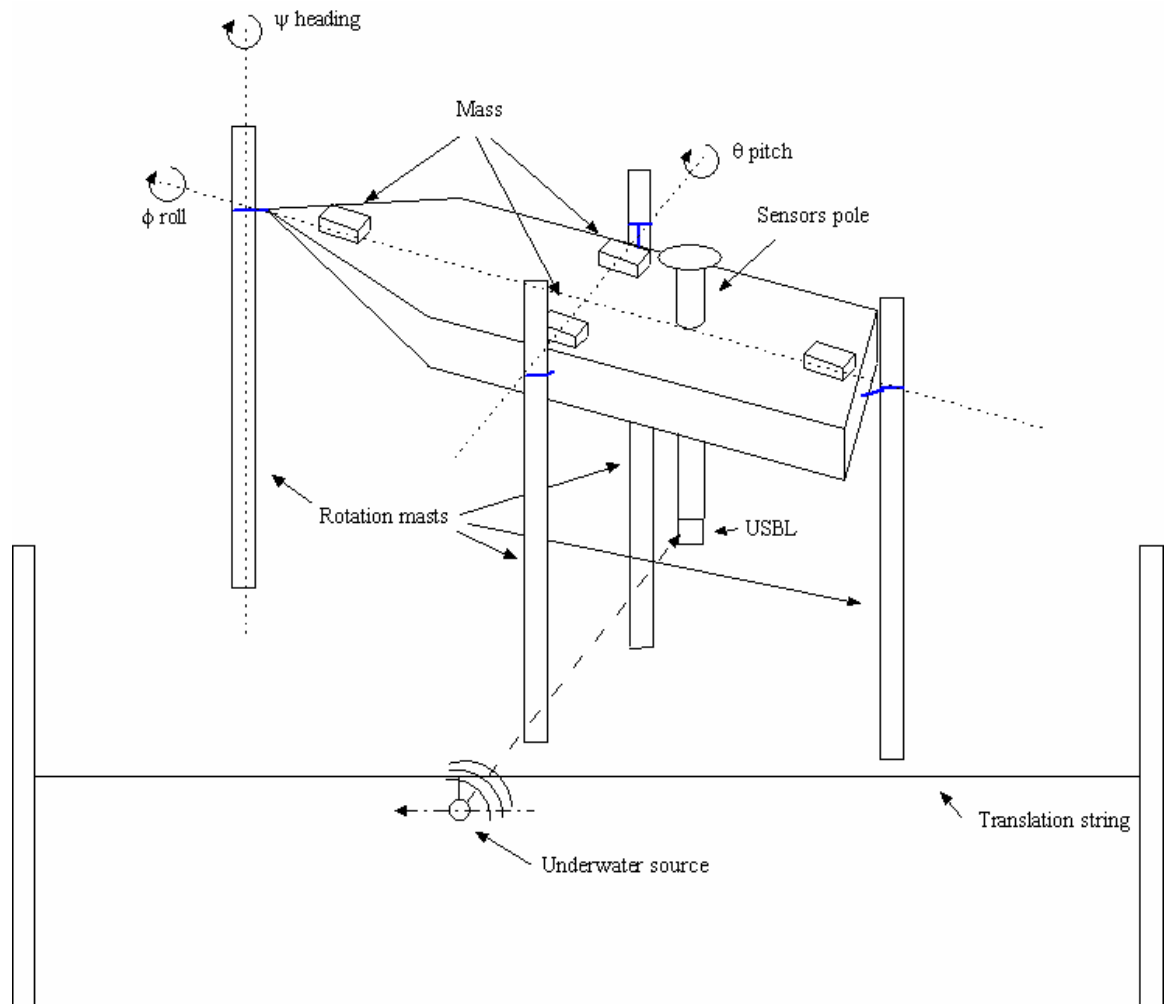


Figure 2.3.15. USBL test setup.

### 2.3.4 High-speed acoustic communications:

The high-speed high-frequency acoustic modem technology presented in this section has been developed in part under separate funding from the Office of Naval Research, Science and Technology (code 32). The resulting financial and technical leverage allowed us to obtain the results shown here.

A one-way, high-speed, high-frequency acoustic modem (HS-HFAM) operating between 260 kHz and 380 kHz has been developed to transmit compressed images from an underwater vehicle to a surface operator [11-16]. High data rates are made possible using a high-resolution decision feedback equalizer with parallel algorithm for tracking and compensating large Doppler, developed at Florida Atlantic University (FAU). Two prototypes have been developed at FAU in partnership with EdgeTech, Inc. The source

units are small (0.5 m in length by 0.12 m in diameter), lightweight and power-efficient. The receiver unit is a small, self-contained, lightweight and splash-proof case combined with a laptop. Small, single channel commercial transducers are used to transmit and receive the broad-band acoustic sequences. The HS-HFAM is a one-way high-speed acoustic modem designed to transmit combined images and text information in very short bursts. The HS-HFAM communication system accommodates multiple inputs and outputs using Ethernet connections. The inputs and outputs are either UDP or TCP-IP ports.

The three main concerns associated with broadband acoustic communication at high frequencies in harbors are reverberation, Doppler shift and, to a lesser extent, noise. Acoustic reverberation, originating from the scattering of acoustic waves off the surface, bottom, walls and obstacles, causes inter-symbol interference (ISI). In the frequency domain, reverberation is equivalent to frequency-selective fading. Frequency-selective fading also includes the effect of sound refraction due to sound velocity gradient. Doppler shift is due to the relative motion of the communication platforms and boundaries, especially the water surface ship hulls and some biological life. The combined effect of various Doppler shifts is known as Doppler spread. Background noise due to boat traffic is relatively benign around 300 kHz (approximately 35 dB re  $1\mu\text{Pa}/\sqrt{\text{Hz}}$ ), however thermal noise causes an increase of 6 dB per octave above 100 kHz. The use of high-frequencies for high-speed underwater acoustic communications has significant advantages. First of all, the transducers are small, efficient and can be fitted in small UUVs. Also, the high bandwidth means high data rate and also excellent dual space-time resolution. With this high spatial resolution, DFE (Decision Feedback Equalizing) processes can better compensate the multipath, which is the main cause of limitation of this type of communication devices.

Each message contains three distinct parts used to detect, synchronize, identify and transfer encoded data, while ensuring efficient, error-free reception of the data. The first portion of a message is a 2.7 ms chirp transmitted between 247 and 273 kHz, with a dead-time of 3 ms, and used for detection and synchronization. The second portion is a 5.1 ms message header, which contains the symbol duration (40 $\mu\text{s}$ , 20 $\mu\text{s}$ , 13 $\mu\text{s}$ ) and the type of modulation used (BPSK, QPSK). The data packet is received 3 ms after the message header. The number of information bits is set to 9120 plus 32 CRC bits, coded with BCH(15,11,1). The message starts with a 512-bit training sequence. The actual packet duration varies from 91.5 ms to 549.1 ms depending on the modulation. The true information bit rate varies from 16243 bps to 87768 bps. A tone is transmitted simultaneously at 375 kHz for efficient Doppler tracking. The HS-HFAM is remarkably power efficient: at full acoustic power and at the fastest bit rate, 13298.2 bits of information are transmitted per 1 Joule of acoustic energy. Table 2.3.1 summarizes the salient characteristics of the data packet.

TABLE I  
HS-HFAM DATA PACKET SPECIFICATIONS

Modulation Type	BPSK	BPSK	BPSK	QPSK	QPSK	QPSK
Symbol Duration	40 $\mu$ s	20 $\mu$ s	13 $\mu$ s	40 $\mu$ s	20 $\mu$ s	13 $\mu$ s
Symbol Bandwidth	25 kHz	50 kHz	75 kHz	25 kHz	50 kHz	75 kHz
Information bits/ frame	1140	1140	1140	1140	1140	1140
Packet duration (ms)	0.5491	0.2745	0.1830	0.2745	0.1373	0.0915
Message duration (s)	0.5615	0.2869	0.1954	0.2869	0.1497	0.1039
Information rate (bps)	16243	31784	46668	31784	60935	87768
Packet coded rate (bps)	25000	50000	75000	50000	100000	150000
Bits-per-Joule (bit/J)	2461.1	4815.8	7070.9	4815.8	9232.6	13298.2

The data are collected using a high-resolution, low-noise acquisition system developed by EdgeTech Inc. in collaboration with FAU (Figure 2.3.16). The acquisition system produces complex base-band signals with a 24-bit resolution. These data are processed with a commercial off-the-shelf PC laptop, connected to the acquisition unit via the Ethernet. Each incoming message is detected, authenticated, equalized and decoded, and the output is relayed to a de-multiplexer which routes relevant information to each application. At present, the applications are the imaging sonar topside display and the vehicle control display.



Figure 2.3.16. HS-HFAM source (left) and receiver (right, courtesy of EdgeTech Inc.).

A series of experiment took place from early February 2007 until mid-March 2007. The source was placed on a kayak and moved to various locations, at a maximum range of 118 m. The source transducer was kept as far as possible from the water surface to minimize the pressure release impact. When the water depth was very low, the transducer would sit in mid water. Blue circles mark the various source locations in Figure 2.3.17. Whenever the source was close to the dock, live Didson images were transmitted, otherwise canned Didson images were transmitted. Within a message, 8000 bits were allocated for the image and 928 bits were allocated for other sensor information. The source speed varied between 0 and 0.5 m/s, the water depth varied from 0.5 m to 3 m depending on tide and location. Figure 2.3.17 also shows the bathymetry measured at low tide of the experimental area. The receiver was located between 1 and 1.5 m below the water surface, along the FAU research vessel Oceaneer IV. The receiver location is shown as a red circle in Figure 2.3.17.

Messages were transmitted at a rate of two per second, using full power and the 6 modulations. A typical mission would last 5 to 6 hours. Each battery pack would allow for approximately 18 hours of continuous operations. The bottom type was mud and sand, and the brackish water characteristics were very similar to these of South Florida

coastal waters, due to the proximity of Port Everglades inlet. There was no control on boat traffic or depth sounders, and numerous schools of fish were present.

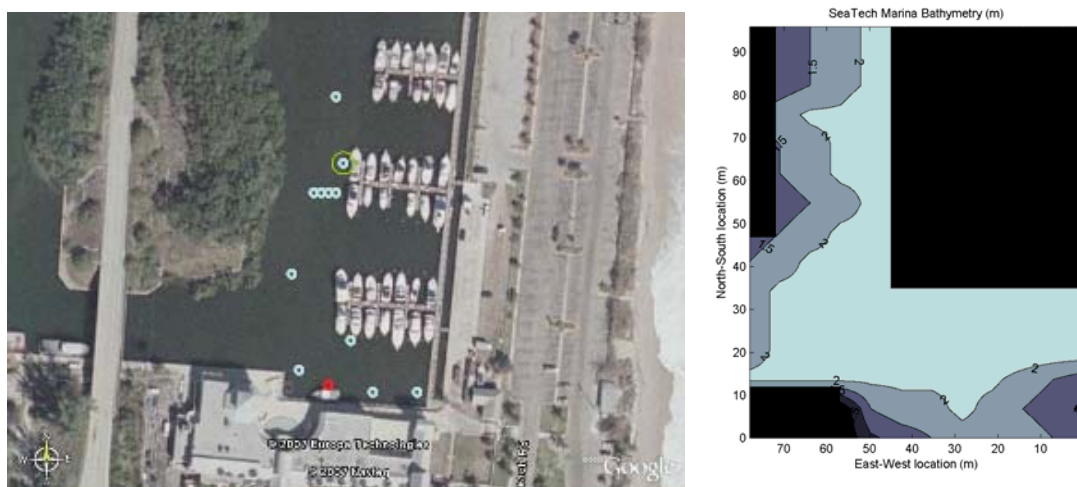


Figure 2.3.17. Experiment overview, SeaTech Marina, Port Everglades, Florida.

A process log for each detected message was stored. Among other information, this log contained the estimate of the average Doppler shift observed in a packet, the impulse response of the acoustic channel estimated from the training sequence, the minimum mean-squared estimation error of the message using the best performing equalizer, and the instantaneous bit error rate for each packet. Table 2.3.2 shows a performance summary of the communication system at the same location using six different transmission modes. The data were collected on March 21<sup>st</sup> 2007, from 1 pm until 5 pm. Overall, the bit error rate remains well within 4% at 88 m, estimated over a total of 15578752 information bits transmitted.

TABLE II  
HS-HFAM PERFORMANCE SUMMARY AT 88 M

Modulation Type	BPSK	BPSK	BPSK	QPSK	QPSK	QPSK
Symbol bandwidth (kHz)	25	50	75	25	50	75
No. of messages	324	217	379	217	425	303
No. of data bits received	2954880	1777664	3104768	1777664	3481600	2482176
Mean BER (%)	3.85%	3.93%	0.84%	3.93%	3.36%	3.01%

## References

- [1] Center for Coastline Security Technology, "Year One: Final Technical Report," Florida Atlantic University and US ONR, June, 2006.
- [2] R. Uhrich and J. Walton, "Supervisory Control of Untethered Undersea Systems: A New Paradigm Verified," presented at The 9th International Symposium on Unmanned Untethered Submersible Technology, Sept. 1995.
- [3] H. Robinson and A. Keary, "Remote Control of Unmanned Undersea Vehicles," presented at the International Unmanned Undersea Vehicle Symposium, April 2000.
- [4] M. Dunbabin, J. Roberts, K. Usher, G. Winstanley and P. Corke, "A Hybrid AUV Design for Shallow Water Reef Navigation," presented at the International Conference on Robotics and Automation, April 2005.

- [5] A. M. Polsenberg, "Developing an AUV Manual Remote Control System," M. S. Thesis, Massachusetts Institute of Technology, 2000.
- [6] P. P. J. Beaujean, "Short Course on Underwater Acoustic Communications", 151st Meeting of the Acoustical Society of America, 5-9 June 06, Providence, Rhode Island.
- [7] P.P.J. Beaujean and E.P. Bernault, "A New Multi-Channel Spatial Diversity Technique for Long Range Acoustic Communications in Shallow Water", MTS/IEEE Oceans'03 Proc., San Diego, CA, Sept. 2003.
- [8] P.P.J. Beaujean, A.I. Mohamed and R. Warin, "Maximum Likelihood Estimates of a Spread-Spectrum Source Position using a Tetrahedral Ultra-Short Baseline Array", Proceedings of OES/IEEE Oceans'05 Europe, June 2005, Brest, France.
- [9] P.P.J. Beaujean, A.I. Mohamed and R. Warin, "Acoustic Positioning using a Tetrahedral Ultra-Short Baseline Array of an Acoustic Modem Source Transmitting Frequency-Hopped Sequences", Journal of Acoustical Society of America, J. Acoust. Soc. Am. 121, 144 (2007).
- [10] R. G. Brown, P. Y.C Hwang, "Introduction to Random Signals and Applied Kalman Filtering", Third Edition", John Wiley & Sons.
- [11] P.P. Beaujean, T. Nguyen, "High-Speed, High-Frequency Acoustic Modem (HS-HFAM) coupled with a Wavelet-Based Embedded Compression", ONR Joint Review of Unmanned Systems Technology Development, Panama City, Florida, Feb. 2006.
- [12] P.P. Beaujean, "High Data Rate, Short Range Communications", ONR Joint Review of Unmanned Systems Technology Development, Panama City, Florida, Feb. 2005.
- [13] J. Spruance, P.P. Beaujean, "A High-Speed High-Frequency Broadband Acoustic Modem for Short-to-Medium Range Data Transmission in Ports, Very Shallow Waters and Deep Waters using Spread-Spectrum Modulation and Decision Feedback Equalizing", Phase-I STTR, final report, March 2006.
- [14] J. Spruance, P.P. Beaujean, "A High-Speed High-Frequency Broadband Acoustic Modem for Short-to-Medium Range Data Transmission in Ports, Very Shallow Waters and Deep Waters using Spread-Spectrum Modulation and Decision Feedback Equalizing", Phase-I STTR, 2<sup>nd</sup> status report, December 2005.
- [15] J. Spruance, P.P. Beaujean, "A High-Speed High-Frequency Broadband Acoustic Modem for Short-to-Medium Range Data Transmission in Ports, Very Shallow Waters and Deep Waters using Spread-Spectrum Modulation and Decision Feedback Equalizing", Phase-I STTR, 1<sup>st</sup> status report, September 2005.
- [16] P.P.J. Beaujean, P.M. Blue and D. Kriel, "A High-Speed High-Frequency Acoustic Modem (HS-HFAM) for Ports and Shallow Water Operation", 13th International Congress on Sound and Vibrations, Vienna, Austria, July 2006.

## **2.4 Environmental Assessment and Modeling: Monitoring Turbidity in Ports**

**PI: Dr. George V. Frisk**

### **Tasks 3.8-3.11**

#### **2.4.1 Background**

##### **2.4.1.1 Turbidity**

Turbidity is a measurement of the concentration of suspended particles in a liquid. In port environments, the suspended particles are most often comprised of stirred sediment from the bottom, phytoplankton, and dissolved organic matter, all suspended in an estuarine water type [1],[2]. Turbidity is commonly measured in either Nephelometric or formazine turbidity units. These units are not directly proportional to the concentration, albedo, absorption, or scatterance of the suspended particles; they are relational values of scatterance at the angle used by a turbidity meter to that of a calibration solution, commonly formazine. However, turbidity does not take into account the absorption and attenuation of the water, which will affect the apparent scattering.

Turbidity plays a key role in limiting the effectiveness of optical imaging systems in harbor environments. This can be affected by a variety of natural and human forces, including tidal currents, winds, waves, shipping, dredging, and biological activity.

##### **2.4.1.2 Turbidity Measurement**

In a turbid environment, light will be periodically scattered off particles. By emitting light into the water and detecting it at a backwards acute angle, an approximation of the scattering can be obtained and loosely related to particle concentration. Often times the angle of detection is centered at a 120 degree offset from the light source, which is the angle of minimum variation in the phase scattering function [3]. The angular detection limit usually encompasses a wide range around its center in order to reduce the error caused by the variety of phase scattering functions that the particles may have. Absorption of the light in the water will decrease its luminosity at the detector, which will be falsely interpreted as reduced scattering, and thus reduced turbidity. In order to minimize this, the frequency of light is often chosen at a common minimum of absorption by common ocean water constituents somewhere in the infrared region.

All turbidity measurements were and will be made using the Seapoint Turbidity Meter, which is shown in Figure 2.4.1 and Figure 2.4.2. This is a small, rugged and easy to attach instrument that can be attached to a conductivity, temperature and depth (CTD) instrument to allow for simultaneous measurements of salinity, temperature, and sound speed, in addition to turbidity. The instrument's sensing volume extends only 5 cm from its viewing windows, which is advantageous to us as the Port can contain tight spaces and obstacles. The light source used operates at 880 nm, which is the extreme minimum of

absorption from most types of phytoplankton and chromophic dissolved organic matter (CDOM) [2]. Pure water has a high absorbance at this frequency, but this absorption is known and nearly constant, enabling software to reduce its effect in the final turbidity measurement.

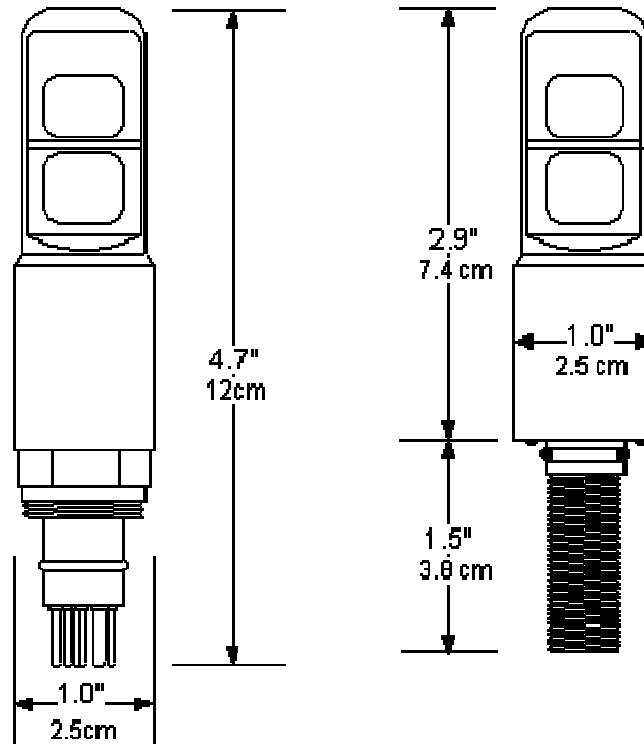
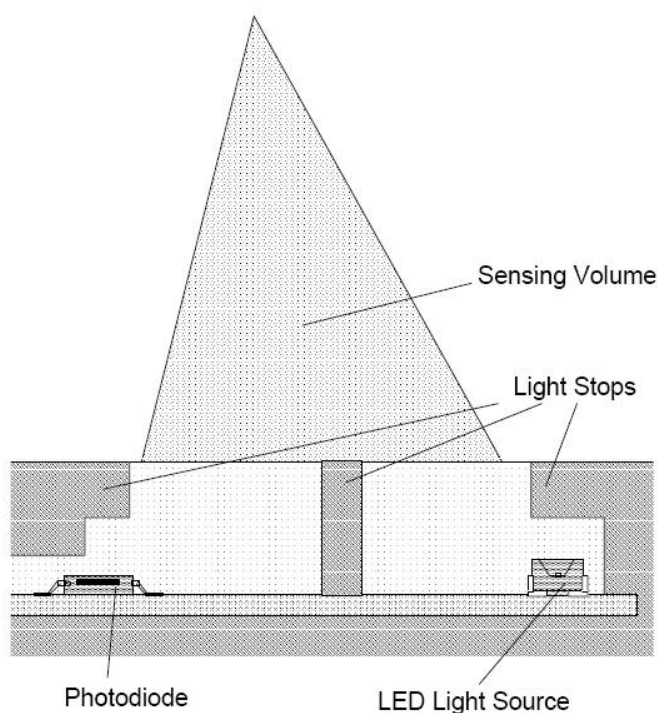


Figure 2.4. 1: Seapoint Turbidity Meter



**Figure 2.4. 2: Cross Section of Seapoint Turbidity Meter Optical Components**

#### **2.4.1.3 Advantages of Turbidity Measurement**

Turbidity measurements were one of the very first optical measurements made in the field of ocean optics. They use very simple and cheap equipment, and in situ measurements are quick and easy. They are especially useful in determining the concentration of suspended particles in solutions where the properties of the particles and liquid are well known. Also, they can give excellent insight into large scale variations in the optical properties of the ocean, and provide a good starting point that can be used to determine further optical studies of the water.

#### **2.4.1.4 Disadvantages of Turbidity Measurement**

The inherent optical properties of the water cannot be derived from turbidity measurements alone. The turbidity measurements can be easily altered due to changes in the size distribution and phase scattering coefficient of the particles. Absorption will also introduce significant error unless it is accounted for, which turbidity meters do not do. Turbidity measurements are made at only one wavelength which may not be the frequency of interest when applying these measurements.



## **2.4.2 Methodology**

### **2.4.2.1 Port Everglades Turbidity Measurement**

Although there have been many previous measurements of acoustical and optical properties in the surrounding areas offshore, there have been none made within the Port itself. Due to this lack of existing data, a sampling strategy was developed to acquire new information. The work was accomplished using FAU research vessels to conduct fifteen at-sea trips in which more than 180 profiles were taken using a Falmouth Scientific CTD and Seapoint Turbidity Meter. Two casts were taken at each location being profiled to ensure accuracy of results, each of which took approximately fifteen minutes and varied depending on the bathymetry of that region. A list of the equipment used for these measurements is detailed below:

- Falmouth Scientific Instruments (FSI) NXIC CTD ADC.
- Seapoint Sensors Inc. Turbidity meter.
- R/V Oceaneer operated by Florida Atlantic University (FAU).
- R/V Stephan operated by FAU.

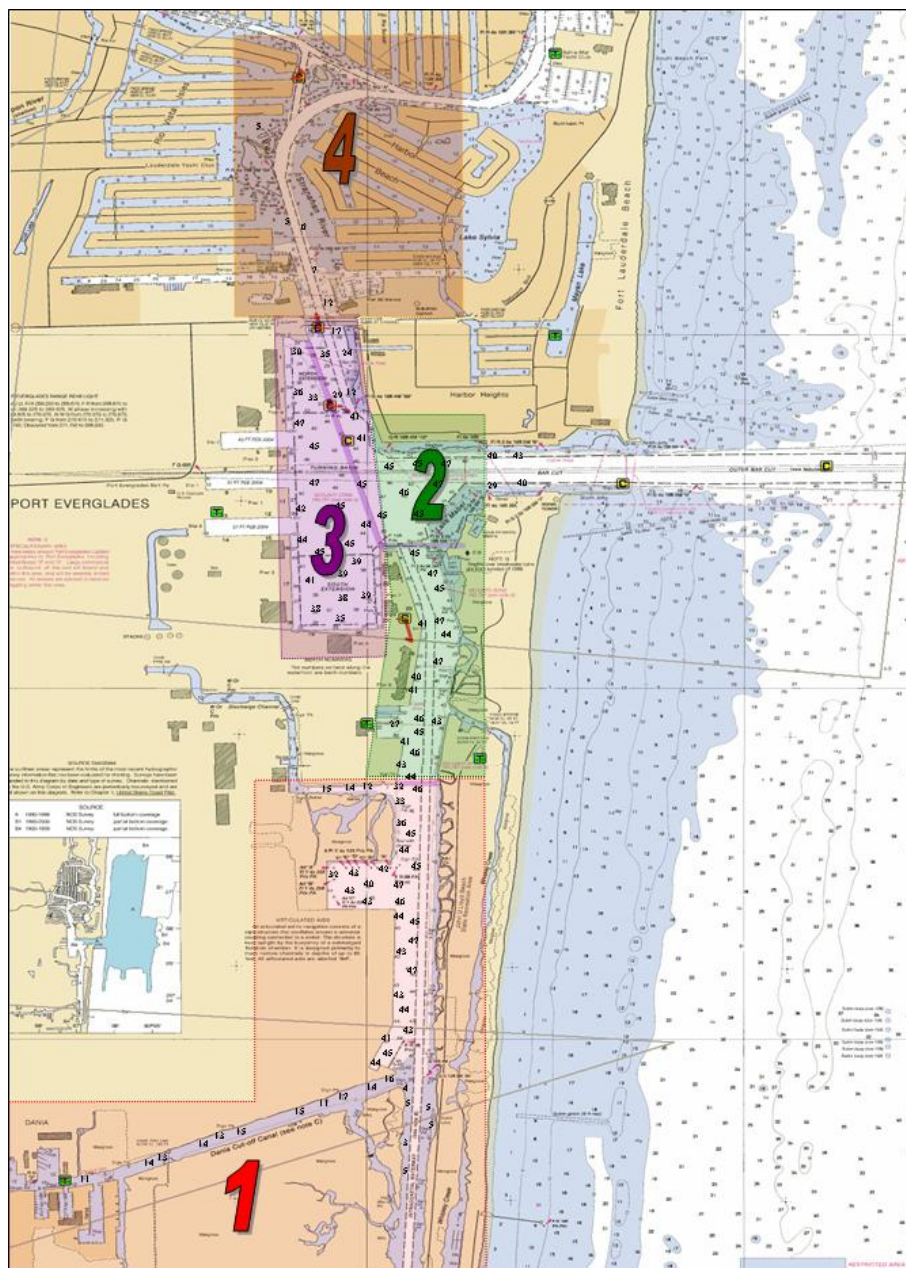
In addition to almost 180 casts made around the Port, measurements were made offshore for comparison purposes. The offshore measurements consistently showed extremely low turbidity values and will be considered as zero henceforth. CTD and turbidity measurements were obtained on the following dates in 2006:

- March 28<sup>th</sup>
- April 5<sup>th</sup>, 7<sup>th</sup>, 12<sup>th</sup>, 21<sup>st</sup>, 26<sup>th</sup>
- May 3<sup>rd</sup>, 4<sup>th</sup>, 5<sup>th</sup>, 8<sup>th</sup>, 9<sup>th</sup>, 10<sup>th</sup>, 11<sup>th</sup>, 17<sup>th</sup>, 18<sup>th</sup>

## **2.4.3 Results**

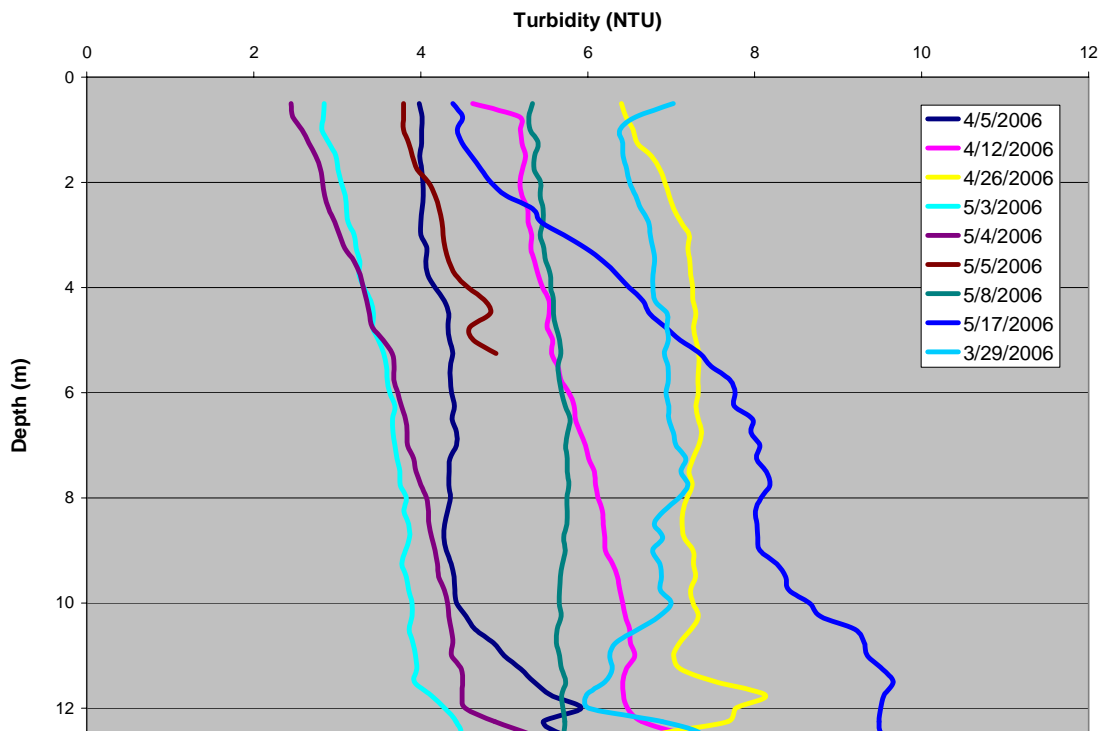
### **2.4.3.1 Temporal Dependence**

After initial measurements were taken, it became clear that the turbidity within Port Everglades is subject to high variability. The first step in analyzing the optical properties would be to gather an understanding of the temporal variability, allowing an understanding of what turbidity range should give a good estimate for predictions on a given day. To aid in this analysis, it was pertinent to approach the Port as an area broken into four regions, each with area specific characteristics. This chart is shown in Figure 2.4.3.



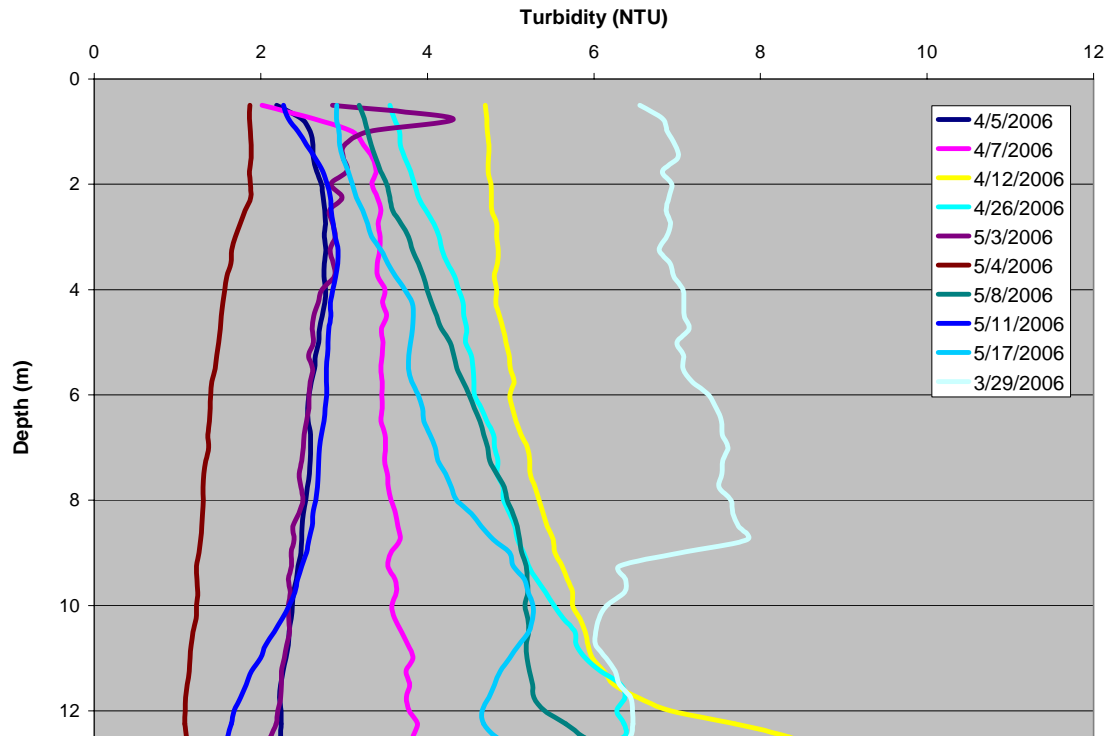
**Figure 2.4. 3: Port Everglades with Component Regions**

The time-dependent profiles in Figures 2.4.4 – 7 each show one turbidity profile for each day that measurements were taken, within each of the four regions. If no measurements within that region were taken on that particular day, there will be no representation of it. If multiple profiles were sampled within the region on a specific day, all the profiles from that day are averaged to create one representative profile, since in general, there is not much variability per region, per day.



**Figure 2.4. 4: Region 1 Turbidity Profiles**

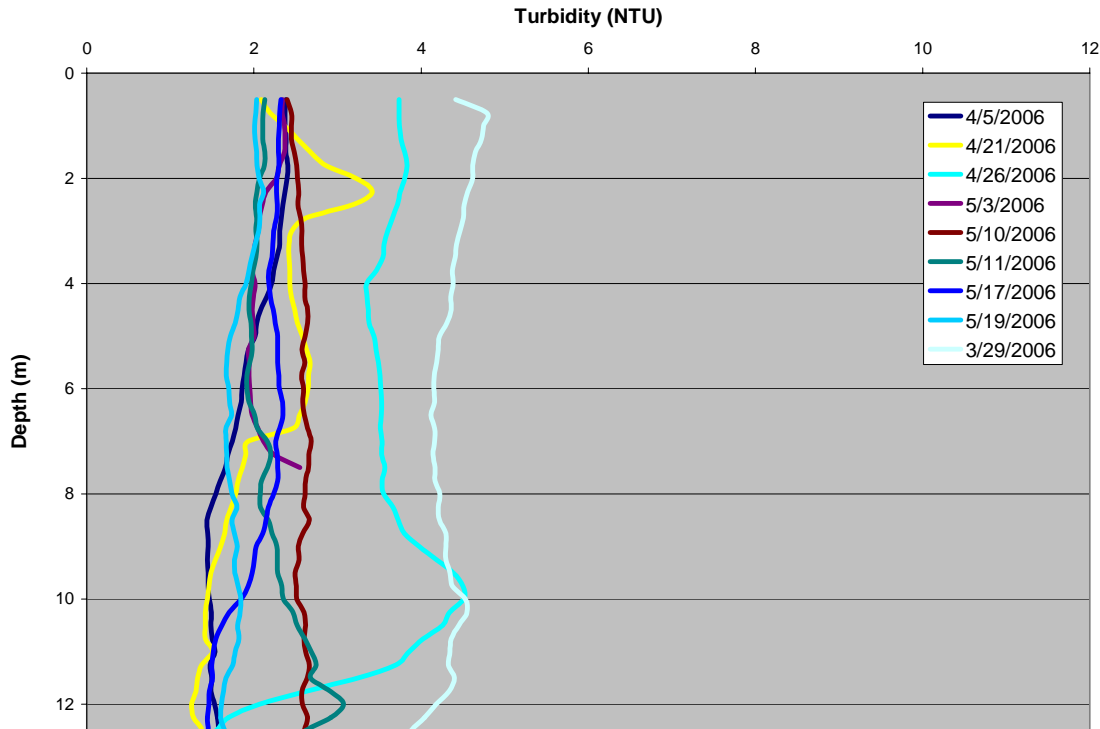
As the graph shows for Region 1, the turbidity varies considerably from day to day. The turbidity usually ranges from approximately 3 NTU to 8 NTU. In this region, most of the profiles show a turbidity increase towards the sea bottom. This is most likely due to the stirring up of sediment from the bottom by passing or docking ships. The highly sloped profiles indicate possible recent ship activity and demonstrate the sediment settling effects. The days with values approaching iso-turbidity are possible indications of the natural turbidity levels when shipping activity is sparse or absent.



**Figure 2.4. 5: Region 2 Turbidity Profiles**

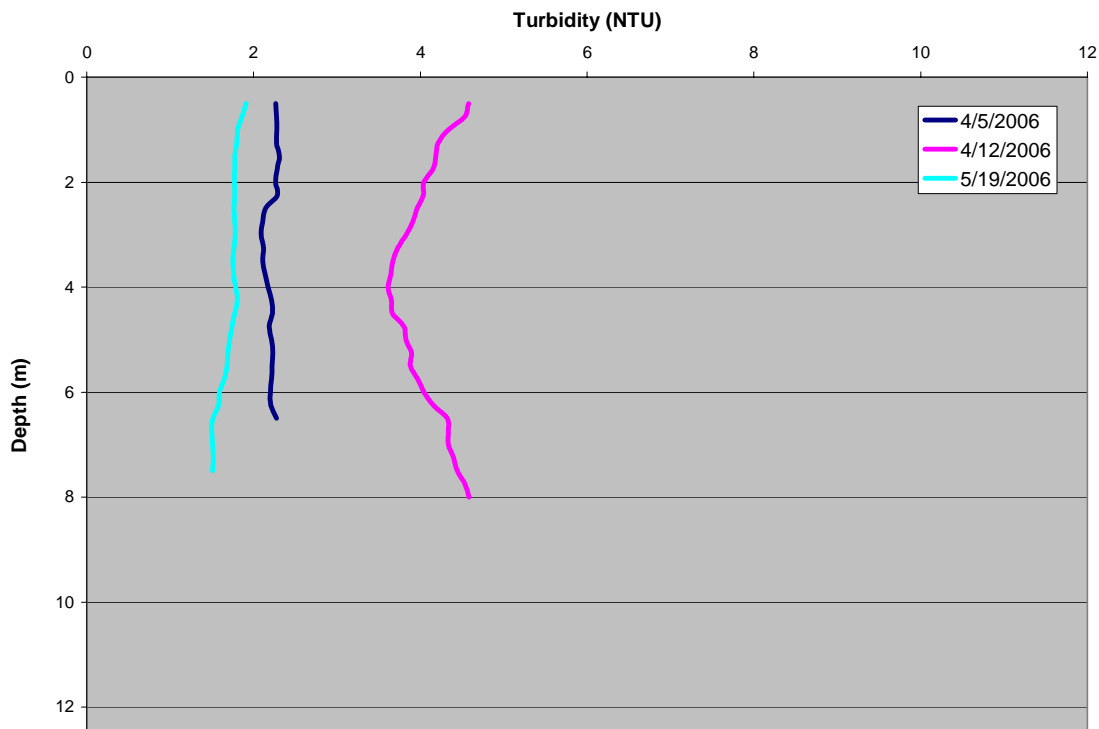
The turbidity within Region 2 generally stays between 1 NTU and 5 NTUs, except for the deepest values on 4-12-2006, and the entire water column on 3-29-06. The lack of significant increases in turbidity towards the bottom (as seen in Region 1) could be due to several reasons. Although some of the water flowing into this region has terrestrial origins, a much greater amount will come from the port inlet and thus the open ocean. As the water moves away from the inlet, the heaviest particles will be deposited first, followed by the lighter sediment later, and further from the port entrance. This would cause the bottom sediment in this region to be heavier and thus exhibit less time in the water column due to its much faster settling time.

Another explanation arises from the possibility that there is not much sediment in this region that is able to be stirred up. The generally lower values of turbidity throughout the water column is expected, also because of the proximity to the port inlet, which means that more ocean water rather than terrestrial water will comprise this region. The profile taken on 3-29-06 which shows high turbidity near the surface and then abruptly drops off below 9 meters is possibly due to phytoplankton occupation. In order for phytoplankton to survive it needs a certain minimum amount of sunlight which can only be found in the top parts of the water column. This profile is a strong indication that the turbidity for the first 9 meters of depth is due to phytoplankton. The turbidity below this depth is most likely due to the same effects as the other days of measurement. The depths to which there is a sufficient amount of light decreases with increased phytoplankton concentration, which in turn increases the apparent turbidity.



**Figure 2.4. 6: Region 3 Turbidity Profiles**

Region 3 comprises the Port turning basin and should receive significant influx of ocean water that has had only a minimal amount of mixing with the turbid Port water. The lower values of turbidity throughout the profile indicate this clearer water. The lack of depth dependent variations in this region hints that the water here is highly mixed and should exhibit the same optical properties throughout the water column, although this is an estimation. The turbidity between different depths could be caused by different waterborne constituents giving rise to variations in the spectral inherent optical properties here. In order for this to be the case, the effects of the different constituents on the turbidity meter will need to effectively cancel each other out either from changes in absorption, or the phase scattering functions of the particles. In order for these to exactly cancel each other out would be highly unlikely, and will be disregarded henceforth. Turbidity variations between the days could very well be caused by completely different constituents, not just their concentrations. This case will be looked into further with spectral measurements of the attenuation and absorption coefficients of the water.



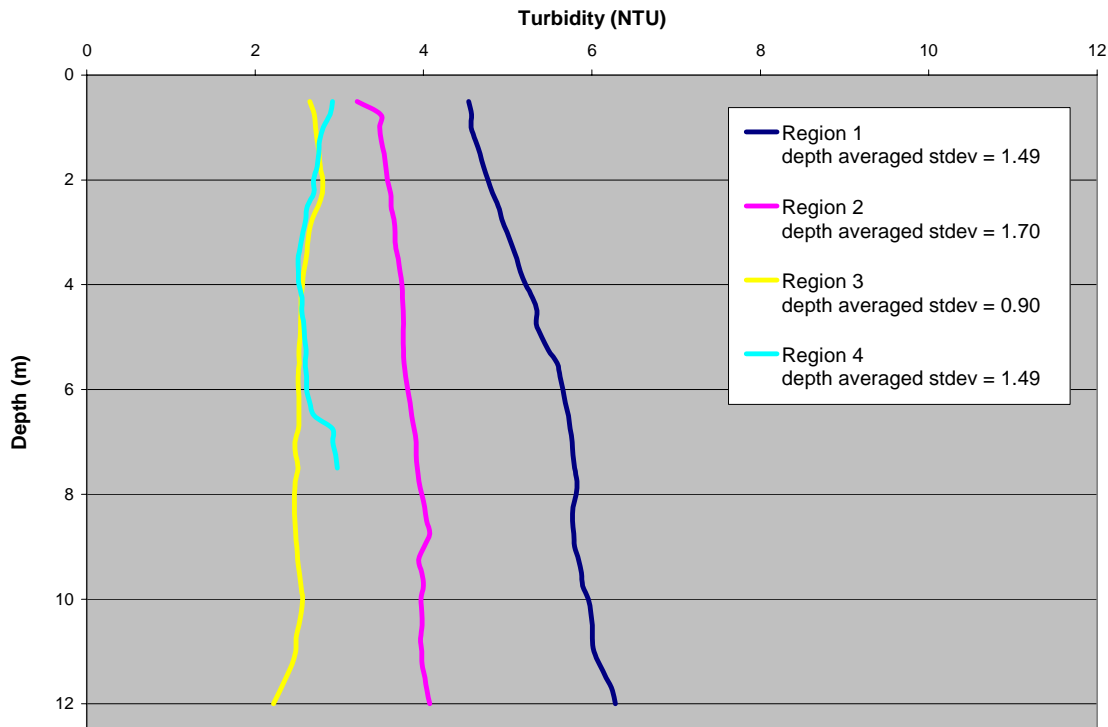
**Figure 2.4. 7: Region 4 Turbidity Profiles**

Region 4 encompasses the port north of the 17<sup>th</sup> Street Bridge. This section is designed for private vessels with a shallow draft, and not for any of the large commercial ships or tankers. This region has a channel running through the center of it which can be seen on the regional map of the Port. The channel maintains an approximate depth of 7 meters while the bathymetry outside of the channel can range anywhere from 1 to 3 meters. Because of the paucity of use of this section of the port only a minimum amount of data was collected in this region. From our measurements it seems that the turbidity here ranges from approximately 1 NTU to 5 NTU. Since these measurements were taken within the channel, a high amount of mixing should be expected due to the boat traffic as well as the draft to depth ratio of many of the ships which frequent the channel. Further studies into this region will offer clues as to the causes of this turbidity behavior.

### 2.4.3.2 Spatial Dependence

The previous graphs provide a good view of the variability of each region on a daily basis as well as a good visualization of its range of variation. For the basis of our project, it would also be to our advantage to compare the regions with one another. An average for each region was calculated for this analysis using the daily averaged profiles instead of all the profiles. This averaging negated weighting effects dependent on number of measurements per day, and gives a better average for spatial dependence.

Comparison of these averages together on the same graph will provide some indication as to the differences between the regions averaged over a two month period. This can be useful for determining which part of the port would be best on average to deploy underwater systems that rely on visual clarity. The depth-averaged standard deviation has also been calculated to show the relative variability between the regions.

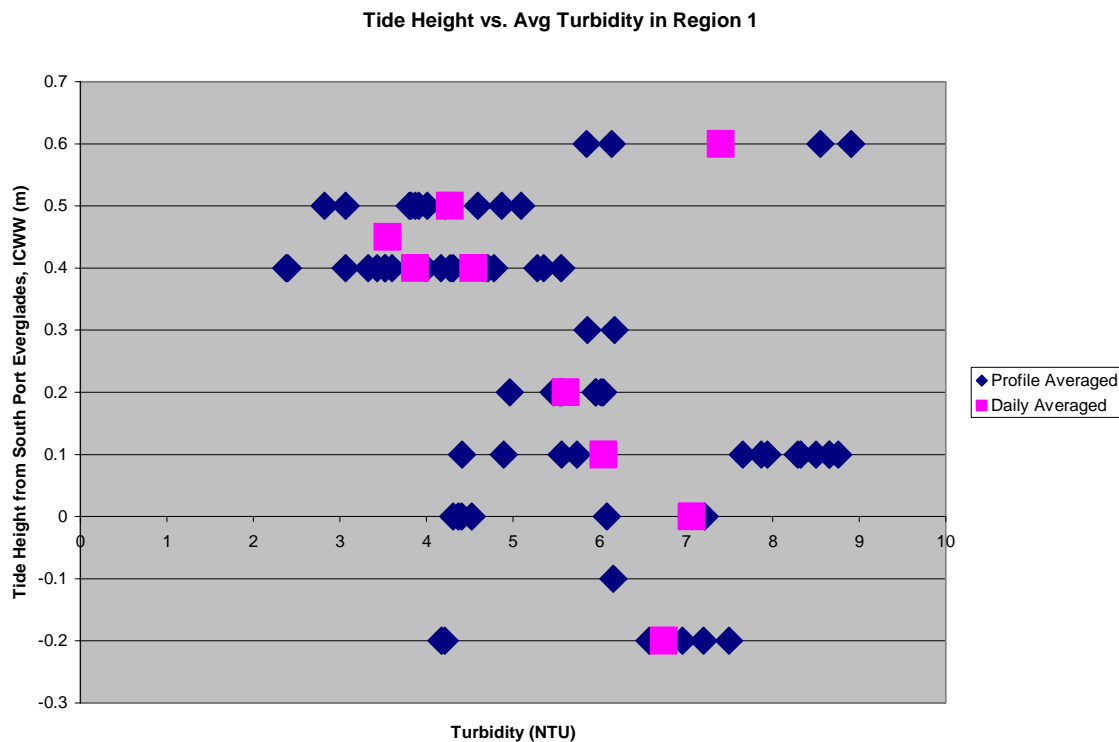


**Figure 2.4. 8: Spatial Turbidity Profiles by Region**

The noticeable feature of this graph is the increase in turbidity by region as the region moves farther away from the Port inlet. Also of note is the subtle increase in slope of the profiles as the region moves farther from the inlet. This illustrates the change in the particle settling properties as the region moves away from the inlet. Also of note is the lack of any features that would indicate turbidity caused by the presence of phytoplankton. This does not mean that phytoplankton does not periodically significantly affect the optical properties of the Port, but it seems that it is not a common feature. Despite Region 3 and 4 being approximately the same distance from the port inlet they show quite different turbidity profiles. This is most likely due to the incoming waters tendencies to flow north rather than south. This explanation could be justified with current measurements around the Port which will be reviewed later.

### 2.4.3.3 Turbidity Correlation

Turbidity within an estuarine environment tends to be higher than that of the open ocean for several reasons. For one, the water column is shallow relative to the open ocean which means that surface effects will tend to disturb the bottom. Also, the water affected by the stirred up sediment will represent a higher percentage of the water column. Because of the relative clarity of ocean to estuarine water, tidal flow will often have a large affect on turbidity. A drop in turbidity should be expected during high tide owing to an influx of clearer ocean water, and a subsequent rise in turbidity during low tide due an influx of terrestrial water. Using data from sensors controlled by Port Everglades, the tide and current data can be assessed at different locations and correlated to the turbidity profiles taken during this project. A chart for each region has been created which shows the relationship that the tide has on the depth-averaged turbidity for each profile. Data points based on the turbidity average over each day's sampling period, usually 1 to 2 hours, is also presented to see the turbidity variations by day.



**Figure 2.4. 9: Tide Height vs. Average Turbidity for Region 1**



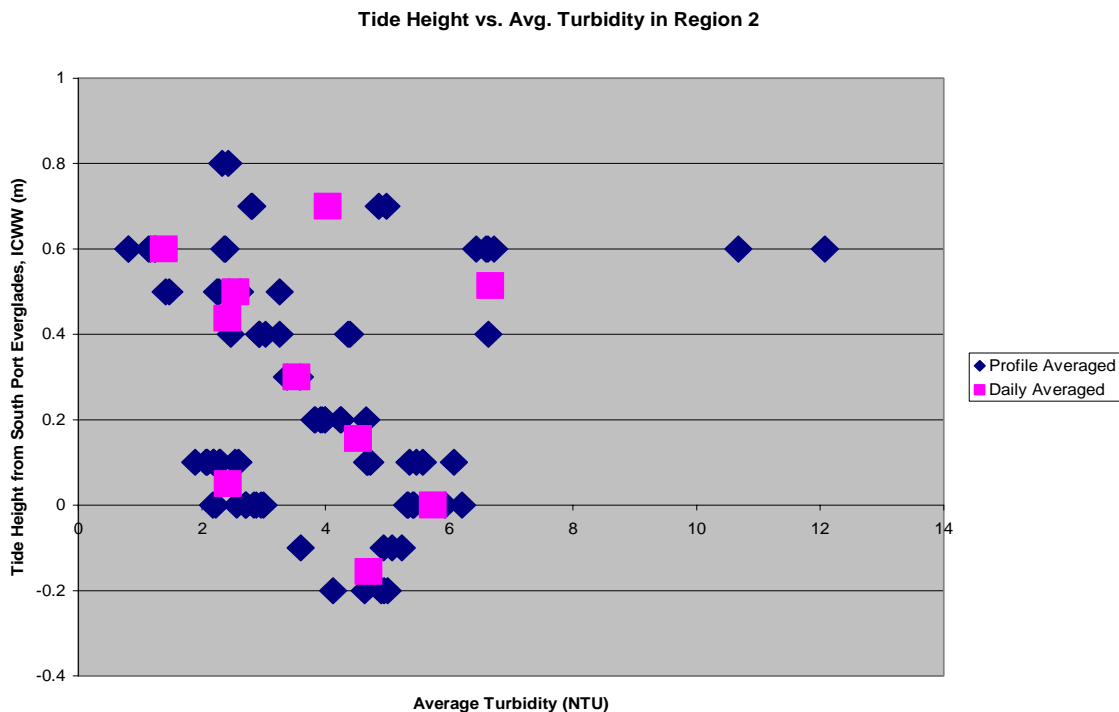


Figure 2.4. 10: Tide Height vs. Average Turbidity for Region 2

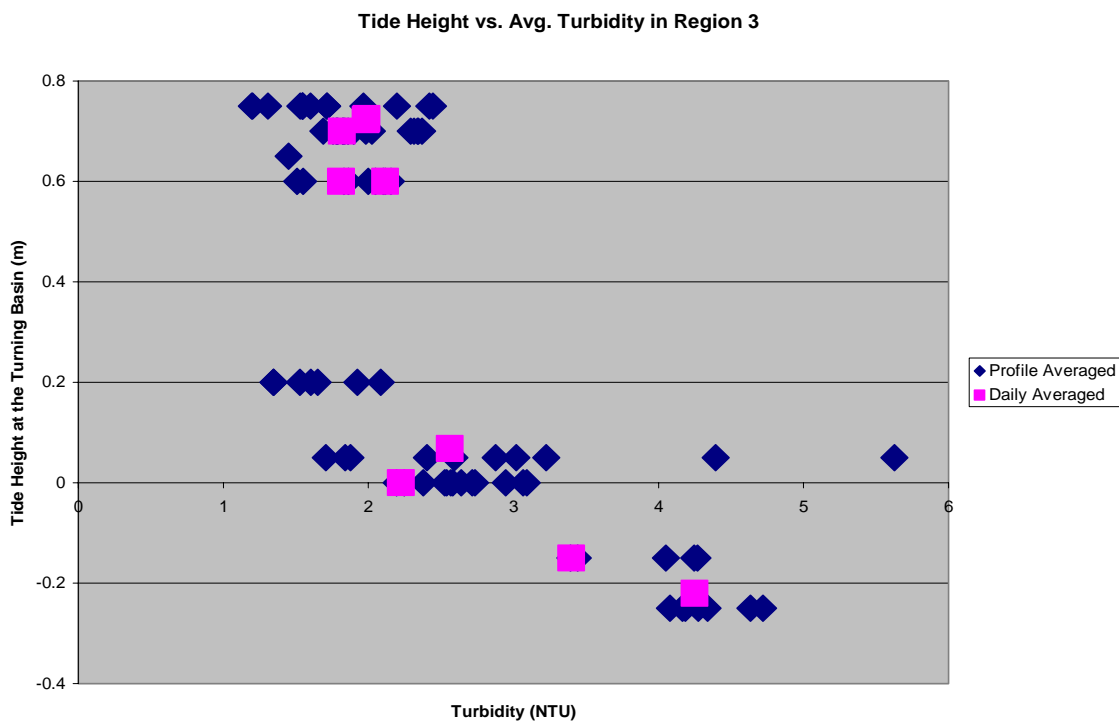
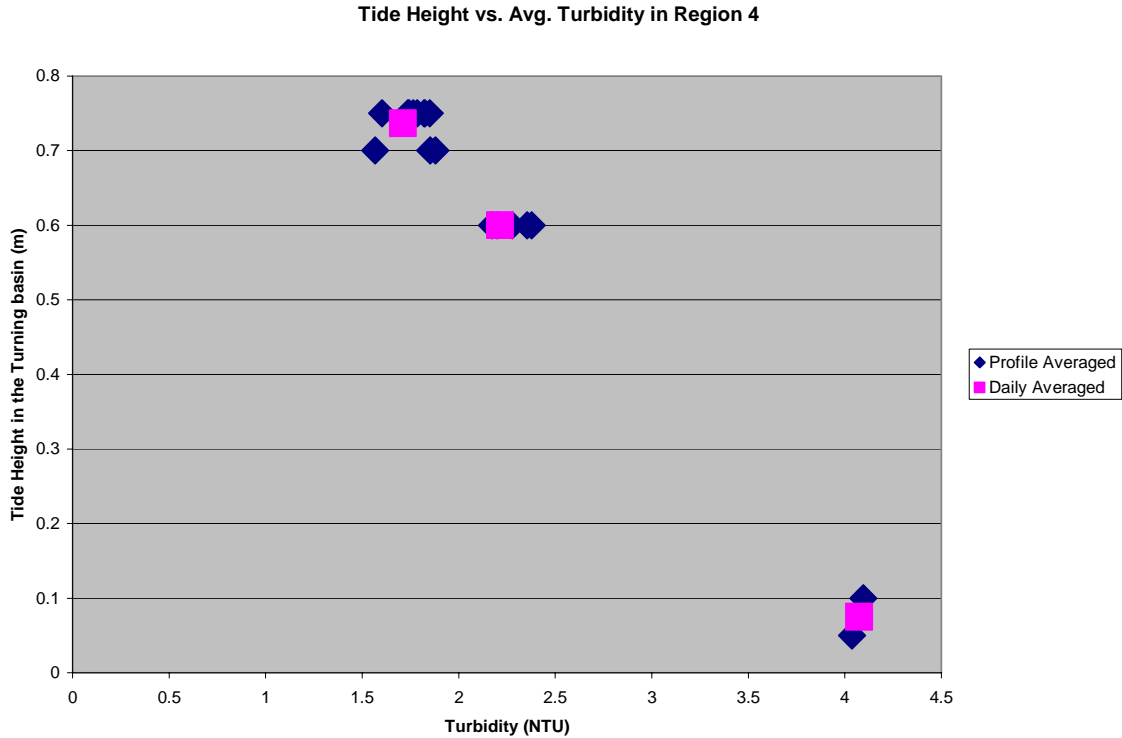


Figure 2.4. 11: Tide Height vs. Average Turbidity for Region 3



**Figure 2.4. 12: Tide Height vs. Average Turbidity for Region 4**

The charts show a slight proportionality of turbidity to tide height. In general, a higher tidal level shows a lower turbidity. The large amount of variation in these data is due to the many other factors that are affecting the port's turbidity at any given time including, port traffic, freshwater flow, and biological activity. Also, although the measurements are represented in their corresponding regions, there are some turbidity variations within each region. All of these other variables combined will yield a wide spread from any easily definable curve.

Of special note is the relatively small correspondence between tide and turbidity within Region 4. This is most likely due to less variance in shipping activity; however there could be other causes. While this section of the port does not receive shipping traffic from large trans-Atlantic vessels it receives a very steady flow of smaller recreational craft. This consistency of boat activity compared to the other regions of the port has kept this effect steady. The subsequent correlation of tide to turbidity in the absence of variability in the shipping activity offers some insight into the relative importance of each to the total turbidity at any point. Information on the ship traffic history in conjunction with current direction and speed around the Port could be used to factor out the shipping activity variability. However, since much of shipping activity is unpredictable within the Port, with the exception of some of the larger vessels, this effect will not be able to be considered for planning specific RPUUV missions. Thus for the purposes of optical classification of the Port for use by the RPUUV, the variability due to boat traffic should not be removed from the data.

Usually turbidity can be compared to salinity with high correlation. This is because terrestrial waters are the genesis of most of the constituents responsible for optical degradation [5]. Also, terrestrial waters usually have much lower salinity. This means that normal offshore measurements will show that the turbidity increases as the salinity decreases.

The Port environment is made up of both estuarine and ocean water types in various levels and so it would be wise to see if a correlation existed of this type. After analysis it became obvious that the salinity/turbidity relationship either does not exist, or is masked by the variability of the other parameters.

#### **2.4.4 Conclusion and Discussion**

The objectives for this part of the project for year 2 are to develop a methodology and system for monitoring the turbidity levels in the Port Everglades environment, including the identification of a COTS optical system for measuring the temporal and spatial variability of turbidity levels. For this purpose, a Seapoint Turbidity Meter has been chosen and integrated with a Falmouth Scientific CTD for simultaneous water measurements in relation to the turbidity. A methodology for the deployment of the device has been chosen to assure a minimization of error and consistency in the data.

Using this methodology aboard an FAU research vessel, 15 at-sea trips and more than 180 profiles have been collected and analyzed. These measurements have shown a high degree of variability within the Port on a temporal and spatial basis ranging from between 1 and 10 Nephelometric Turbidity Units (NTU). Identification of the suitability of areas around the Port to the operation of devices that rely on optical clarity can be recognized by the separation of the Port into specific regions exhibiting similar turbidity characteristics. As expected, temporal variations showed a high correlation to tidal height; however, no relation was found between turbidity and current, salinity, or rainfall. Future work includes detailed spectral absorption and attenuation measurements to gather information on the constituents contributing to the underwater optical degradation.

### 2.4.5 Future Work

Attenuation can usually mean either beam attenuation or diffuse attenuation. The beam attenuation is the measure of light transmittance along a straight line. This means that photons that are either absorbed or scattered out of the optical path are lost to the beam attenuation. The beam attenuation coefficient is the best quantitative measure of underwater visibility. Another measure of attenuation comes in the form of diffuse attenuation which is calculated slightly differently and is important for determining the amount of natural illumination. Both of these values will drastically affect the performance of an underwater camera. Fortunately, the diffuse attenuation coefficient can be derived from the beam attenuation coefficient. Then, coupled with the near surface irradiance, the underwater light field can be calculated [6].

In order to accurately model light through a body of water, the inherent optical properties (IOPs) as well as the light sources need to be known. For the purposes of our investigation it is required to know the properties of light within the water from two sources: the sun and artificial illumination. The inherent optical properties are those that are not dependent upon the light source that's applied, but rather govern how a photon will radiate through the medium. In order to understand the life of a photon from an artificial light source, it will be required to determine the absorption and scattering coefficients. A photon is absorbed when its energy is converted from electromagnetic resonance into vibrational heat energy. This usually happens when the photon encounters a large particle, or a small particle that is specifically designed to absorb, such as chlorophyll. Scattering occurs when a photon's direction is changed by the collision of a particle. As is the case with real world particles the absorption and scattering will both take place in the same medium, but in different amounts. The main constituents in the water that affect a photons life have been documented and their affects are well known. The linear combination of light lost due to absorption and scattering is known as the attenuation and is represented by the equation:

$$c(\lambda) = a(\lambda) + b$$

Where  $c(\lambda)$  is the attenuation coefficient

$a(\lambda)$  and  $b$  are the absorption and total scattering coefficient respectively.

A proper understanding of the underwater light field (ULF) necessitates the measurement of three main properties; attenuation, absorption, and scattering. Each of these three properties is largely affected by the properties of pure water added to the relative concentrations of three main constituents; chromophoric dissolved organic matter (CDOM), chlorophyll-a, and detritus. It is common practice for the classification of estuarine type waters to include the relative percentages of the contributors of the light field. These values along with the total values of absorption and scattering will give essential insight into the water optical properties that can be related to ports of interest.

In order to understand the nature of the optical degradation, a spectrometer will be used to measure the spectral absorption and attenuation coefficients from future water samples. From these two measurements the scattering coefficient can be derived using the definition of attenuation. Aside from the usefulness of the raw spectral properties of the water, these measurements can also be used to determine the type and concentrations of the optically significant constituents. Simultaneous turbidity measurements will be taken with the absorption and attenuation measurements. This will allow for the correlation of turbidity to these properties which will subsequently add pertinence to the future and past turbidity measurements. A tentative measurement schedule for this will involve 6 boat trips between May and July 2007 in which half are taken during low tide and the other half are taken at high tide. Measurements will always be taken at the same locations in order to minimize the error due to spatial variations. These new measurements will be invaluable to designing a proper lighting and camera system for any device that will require visibility within this environment.

## References

- [1] Jerlov, N. G. 1968. Optical Oceanography. Elsevier Publishing Co.
- [2] Petzold, T. J. 1972. Volume scattering functions for selected ocean waters. SIO Ref. 72-78, Visibility Laboratory, San Diego, California.
- [3] T. Oishi. 1990, "Significant relationship between the backward scattering coefficient of sea water and the scatterance at 120 deg, " *Appl. Opt.* 29, 4658-4665.
- [4] Seapoint Sensors, Inc. Seapoint Turbidity Meter User Manual, Exeter, NH.
- [5] Burton, J.D., and P.S. Liss. 1976. Estuarine Chemistry. Academic Press.
- [6] Kirk, J.T.O. 1981, "Monte Carlo study of the underwater light field in, and the relationship between, optical properties of turbid yellow waters", *Aust. J. Mar. Freshwater Res.* 32: 517-532.

## **2.5 Development of a High Resolution Imaging Sonar For Underwater Inspections**

**PI: Dr. Steven Schock**

**Tasks 3.12-3.18**

### **2.5.1 Summary**

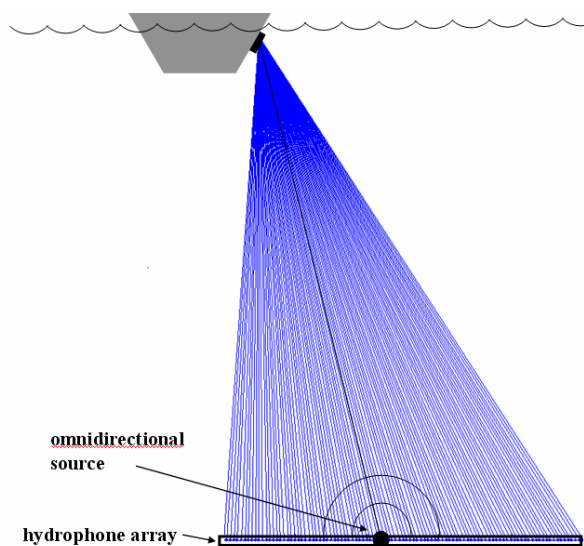
A 256 channel side looking imaging sonar (SLIS) was developed and tested to demonstrate the feasibility of generating a very high resolution acoustic image with a wide field of view (exceeding 90 degrees) using an omnidirectional acoustic source and a discrete line array. The most commonly used commercial acoustic cameras have a field of view of only 29 degrees; such a small spotlight makes it difficult to locate underwater objects using UUVs. However, the SLIS uses a hemispherical source to illuminate a wide field of view providing the sonar with the capability of imaging a wide underwater scene using a single transmission, thereby allowing an acoustic image to be generated on a maneuvering underwater vehicle. This capability represents a major improvement over commonly used acoustic cameras which generate images using several transmissions and thus require the motion of the UUV to be constrained and measured for the purposes of motion compensation during image construction.

Tank measurements of the 256 channel imaging sonar confirmed that the field of view, range resolution and azimuthal resolution agree with simulated image performance. Tank tests were conducted by transmitting an FM pulse over the band of 1.0-1.17 MHz to achieve a half power range resolution of 3 mm. Because the measured half power bandwidth of the 1MHz projector is 340 kHz, this projector provides the sonar with the capability of generating images with a range resolution of 1.5 mm at 1 MHz. For the 25 cm long hydrophone array the azimuthal resolution is a function of range and is equal to 1 wavelength (1.5 mm for a 1.0 MHz pulse) at ranges less than the length of the array. At a range of 4 array lengths, the resolution degrades to 2 wavelengths (3mm for a 1.0 MHz pulse). Because the array length is scalable, a 1 meter long version of the array would generate an image with a 3 mm azimuthal resolution at a range of 3m, a substantial improvement in resolution compared with commercial acoustic cameras. The SLIS generated an image of the test tank walls showing that the sonar range is at least 4 meters.

## 2.5.2 Deliverables

### 2.5.2.1 Acoustic, Electric and Analog Design

The SLIS (side looking imaging sonar) is a 2D imaging sonar designed to create high resolution images with a single transmission. As shown in Figure 2.5.1, an omnidirectional projector illuminates the field of view and backscattering is measured by the hydrophones. The image is constructed by delaying and summing the outputs of the hydrophones for each focal point based on calculations of the ray path from the projector to the focal point and back to each of the receivers.



**Figure 2.5.1 – SLIS (side looking imaging sonar) focusing on an object attached to a hull.**

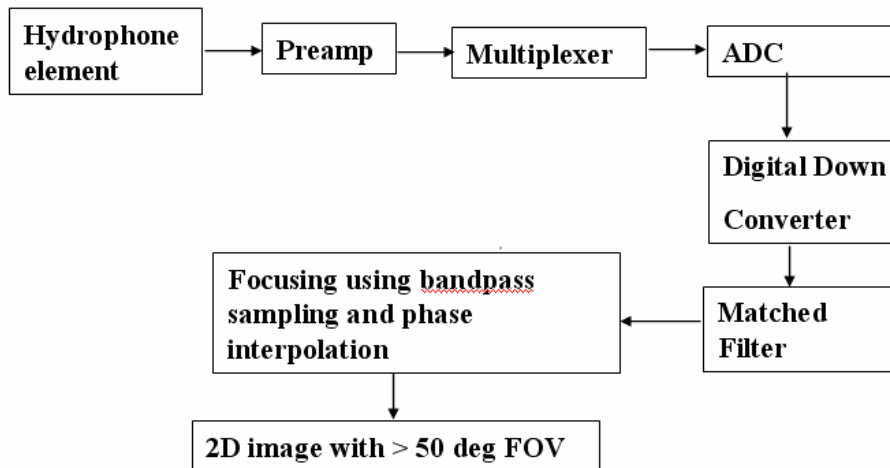
The acoustic, electrical and mechanical design of the SLIS was constrained by the design requirement for 1mm hydrophone element spacing and the desire for a modular design so the length of the array could be easily adapted to the mounting platform. A long array length provides a high azimuthal resolution; however a long array may not be suitable for mounting on a small UUV. The angular location of interfering grating lobes determine the maximum field of view (FOV) because the FOV must be narrower than subtended angle between the grating lobes in order to prevent image artifacts. For 1 mm element spacing, the maximum FOV as a function of operating frequency is given in the following table:

Operating Frequency	FOV
1 MHz	180 degrees
1.5 MHz	128 degrees
2 MHz	84 degrees
3 MHz	52 degrees

The above table shows that the 1mm element spacing will provide a wide FOV up to frequencies of 3 MHz. The actual FOV may be narrower than that given in the above table if the beamwidths of the individual hydrophones are less than the FOV.

The spatial resolution of the imagery is controlled by the bandwidth of the spherical projector, the operating frequency and the length of the array. The bandwidth determines the range resolution. For example, a projector with a bandwidth of 300 kHz will provide a half power range resolution of 1.5 mm. The length of the array and the operating frequency determine the limit of the near field where the center frequency controls the azimuthal resolution. For ranges up to one array length the azimuthal resolution equals one wavelength. Beyond that range, the azimuthal resolution degrades to 2 wavelengths at a range of 4 array lengths and then asymptotically approaches the product of the beamwidth and target range.

The acoustic and electric design supports the signal processing procedures shown in Figure 2.5.2. For the initial tank test of the SLIS, the hydrophone channel outputs were multiplexed prior to analog to digital conversion. In the version of the SLIS to be mounted on the RPUUV, the multiplexing will be performed after analog to digital conversion. Multiplexing reduces the number of power hungry digital down converters needed for signal processing. The digital down converter shifts the high frequency signal to baseband so that the highest frequency component is the bandpass Nyquist rate. The matched filter compresses the FM echoes in time to generate zero phase wavelets. The focusing algorithm executes delay and sum coherent processing and generates a pixel for each focal point in the field of view.

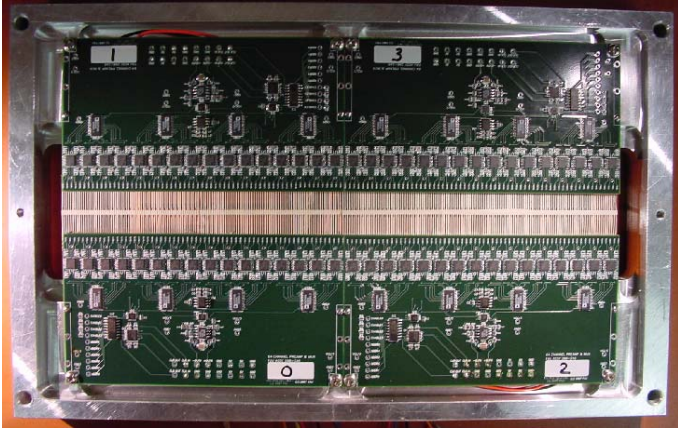


**Figure 2.5.2 – Signal processing flow chart for SLIS**

### 2.5.2.2 Data Acquisition PCB Fabrication

The printed circuits boards performing the data acquisition for SLIS were fabricated, and bench and tank tested. Figure 2.5.3 shows a photo of the assembled preamp and multiplexer PCBs and the PVDF elements just prior to encapsulation during the fabrication of the 256 element receiver array.

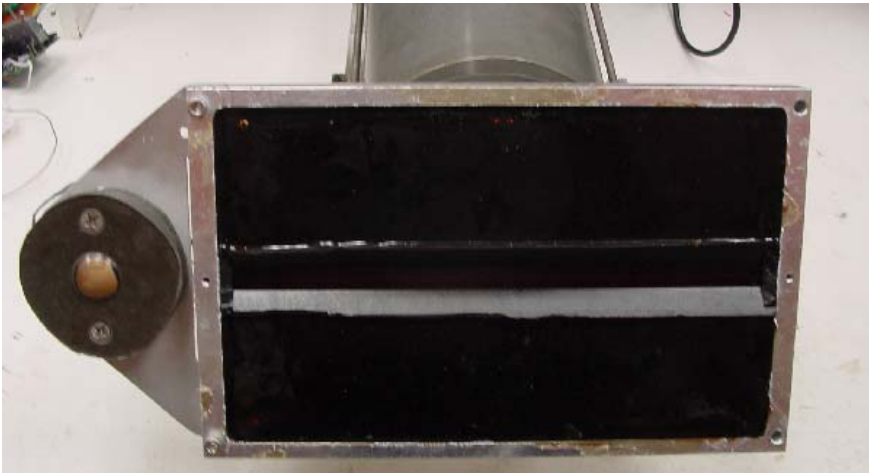




**Figure 2.5.3 – 256 hydrophone elements attached to preamp and multiplexer PCBs**

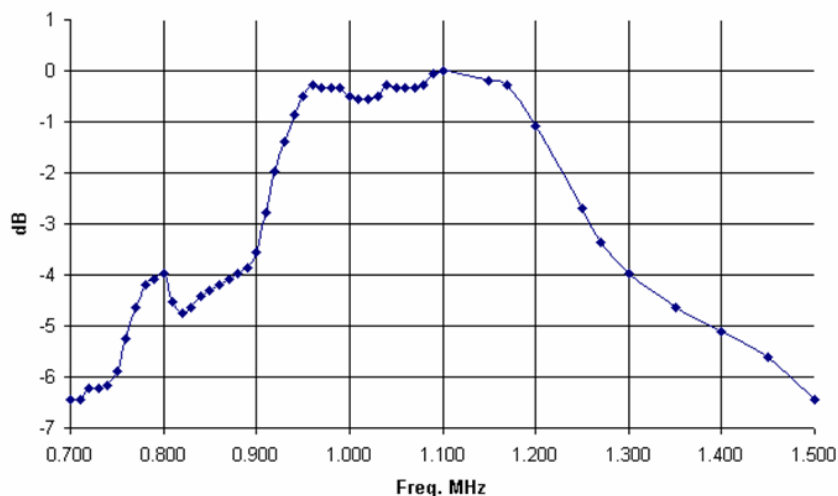
### **2.5.2.3 Hydrophone and transmitter fabrication and performance**

The encapsulated 256 channel hydrophone array and hemispherical transmitter are shown in Figure 2.5.4.



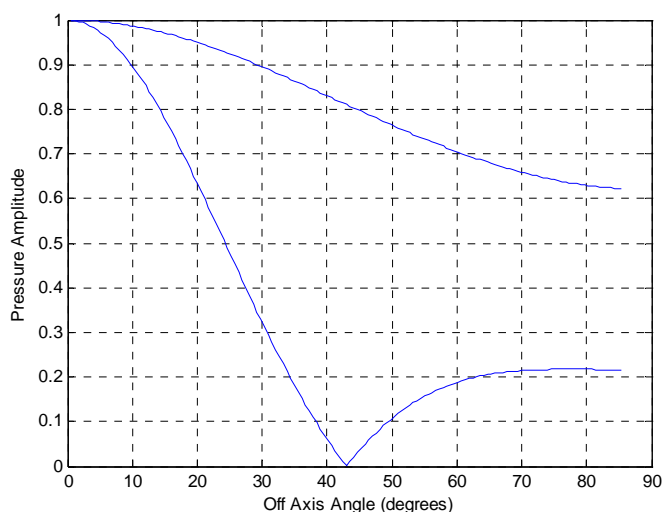
**Figure 2.5.4 – The 256 channel hydrophone array and hemispherical projector mounted on the front the electronics bottle containing the sonar processor.**

The hemispherical PZT projector is approximately 2.5 cm in diameter. The wall thickness determines the resonance frequency of the PZT crystal and thereby the operating frequency of the sonar. The impedance of filler material controls the bandwidth of the projector. A wall thickness of 0.07 inches and a urethane fill provides an operating band of 0.9 to 1.25 MHz as shown by the frequency response measurements reported in Figure 2.5.5.



**Figure 2.5.5 – Transmitting sensitivity of hemispherical projector.**

The expected performance of the 0.7 mm by 2mm PVDF elements is provided in Figure 2.5.6. The  $-3\text{dB}$  horizontal beamwidth which controls the field of view is approximately 120 degrees. Tank measurements showed the FOV was approximately 135 degrees.

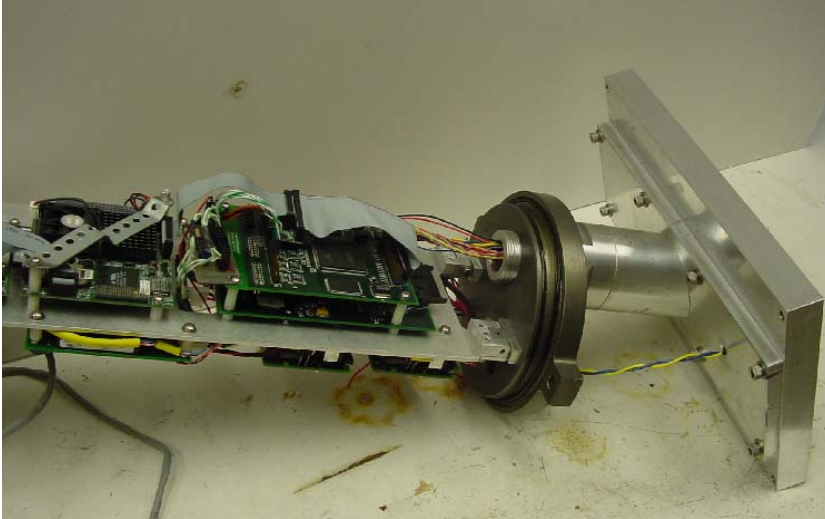


**Figure 2.5.6 The horizontal and vertical beampattern functions for the 0.7 x 2.0 mm PVDF element.**

#### **2.5.2.4 Integration and testing of hydrophone segments sonar processor and data acquisition system on the bench**

Channel to channel phase coherence for the hydrophone array was measured by placing a hydrophone array segment into a small desktop acoustic test tank with the projector transmitting directly into the array segment. There was no measurable phase difference at the outputs of the preamplifiers for the 8 hydrophone channels in the test segment.

The sonar processor, DDC PCB and ADC PCB were packaged in the electronics bottle and connected to the array via a bulkhead connector as shown in Figure 2.5.7. The integrated system passed all bench tests such as measuring noise levels for all 256 channels, searching for digital transmission errors, and exercising sonar control and display software.



**Figure 2.5.7 – The integrated sonar processor and data acquisition system during bench testing procedures**

#### **2.5.2.5 Fabrication of hydrophone array and integrated electronics**

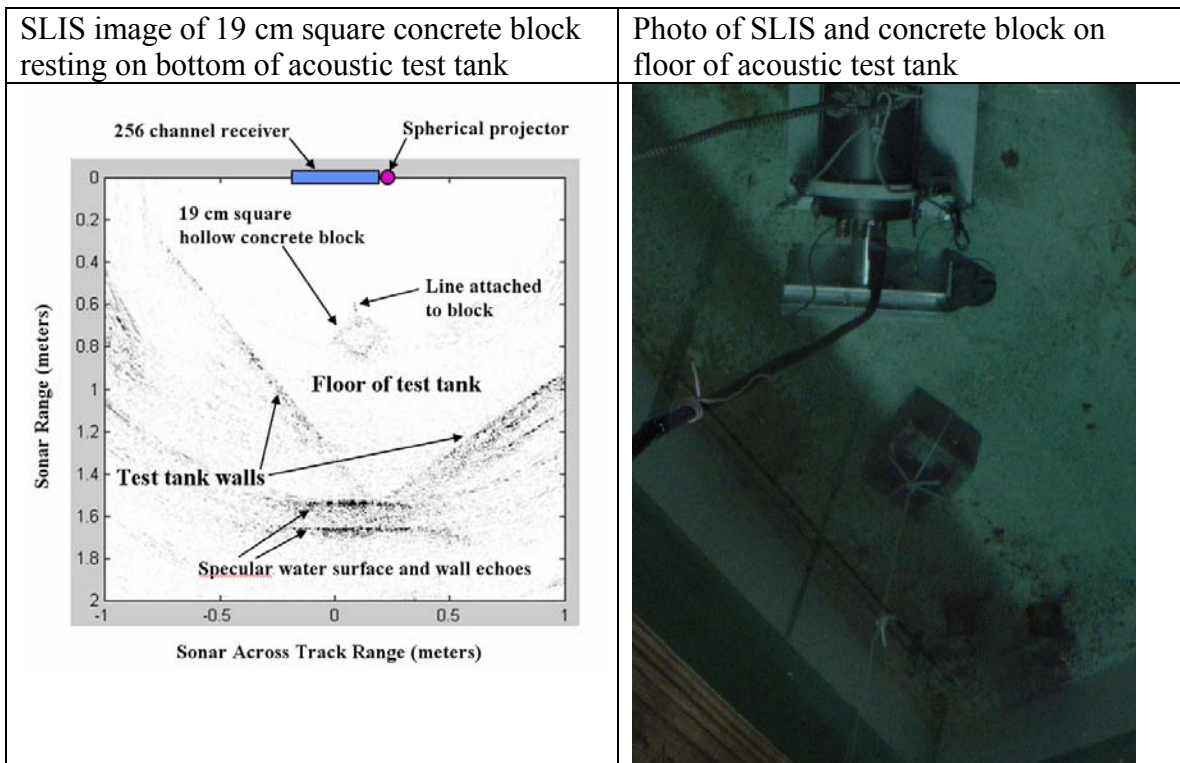
The last step in the fabrication process for the hydrophone array and integrated electronics is to encapsulate the array in urethane. The encapsulated array integrated onto the sonar electronics bottle is shown in Figure 2.5.8.



**Figure 2.5.8 – The 256 channel hydrophone array integrated into the sonar electronics bottle.**

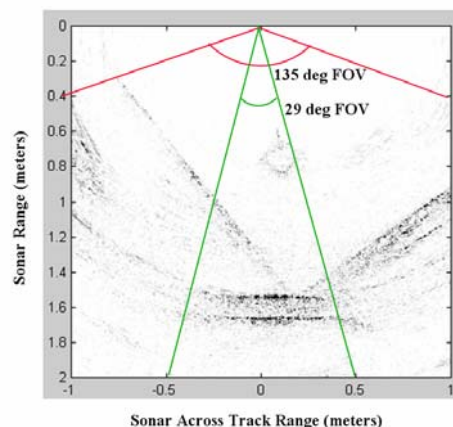
#### **2.5.2.6 Bench and tank tests of the SLIS (side looking imaging sonar)**

Just prior to conducting the tank tests, a final checkout of sonar electronics and software was performed on the bench. For the tank tests the SLIS was placed approximately  $\frac{1}{2}$  meters above the floor of the acoustic test tank and oriented to simulate imaging an object on a smooth surface (such as the smooth hull of a ship). A 19 by 19 cm hollow concrete block as shown in Figure 2.5.9 was placed on the floor of the tank to simulate the threat target attached to the hull. The focused acoustic image clearly shows the concrete block on the tank floor and the side walls of the tank. Water multiples arriving at the bottom of the image emphasize the importance of positioning the 2D SLIS closer to the target than to strong reflectors such as the sea surface or seabed.



**Figure 2.5.9 – SLIS image of concrete block generated during tank testing. Photo of tank test configuration of sonar and target.**

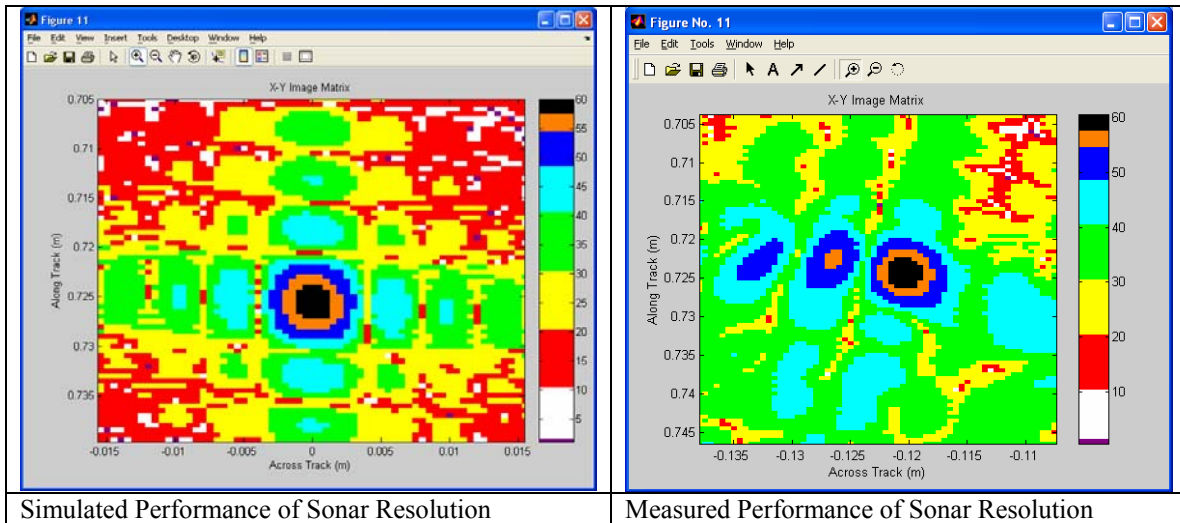
One of the most significant results of the tank tests that demonstrates the capability of the SLIS was the measurement of a 135 degree field of view as shown in the acoustic image given in Figure 2.5.10. This wide FOV is a substantial improvement over the 29 degree field of view of the most widely used commercial cameras.



**Figure 2.5.10 – Acoustic image showing the 135 degree field of view of the SLIS which is a significant improvement over the commercially available acoustic cameras with a 29 degree FOV**



Another significant result of the tank test is the validation of the modeling of the focused acoustic image. The simulated range and azimuthal resolution was compared with the measured range and azimuthal resolution by focusing the SLIS on a polished 4 inch diameter stainless steel sphere. The simulated half power resolutions of 3 by 3 mm agree with the measurements as shown in Figure 2.5.11. Note there is a difference in the sidelobe structure of the focused echo which is due to fact that the simulation was based on a point target and the measurements were made on spherical target with a 4 inch diameter. The curved surface of the spherical target causes some pulse smearing.



**Figure 2.5.11 – Simulated and measured resolution performance of the SLIS at 1.1 MHz using a 256 element, 25 cm long receiver. The amplitude scale of the images is in dB so that black region in the images represents the upper 3 dB of the focused echo peak. The along track width of the black region represents the half power range resolution while the across track width represents the half power azimuthal resolution. Both range and azimuthal half power resolutions are about 3 mm.**

## **2.6 Experimental determination of the hydrodynamic/dynamic characteristics of a small underwater vehicle for port security**

**PI: Karl von Ellenrieder**

### **Tasks 3.19-3.21**

#### **2.6.1 Summary**

The objectives of this research were to study the hydrodynamic design and dynamic response of the RPUUV. This report describes the experimental model, which allows for reconfiguration of the vectored-thruster propulsion system (the control surface of the vehicle), the experimental setup, which permits the model to be tested in various roll, pitch and yaw configurations as well as the experimentally determined hydrodynamic coefficients and thrust output of the vehicle.

Force/torque and particle image velocimetry measurements were conducted on a vectored-thruster UUV model in a water flume/towing tank to: 1) determine the hydrodynamic drag, lift and moment coefficients acting on the vehicle hull for zero rudder angle and yaw angles up to thirty degrees, and 2) measure the magnitude and direction of the thrust produced with the vehicle at a yaw angle of zero degrees and rudder deflection angles of up to thirty degrees. The measured drag coefficient was very close to that predicted by theory; the hydrodynamic coefficients data are expected to be useful in predicting the response of vehicles in the field. Additionally, it was found that the magnitude of the thrust vector varies nonlinearly with rudder angle and for nonzero rudder angles the thrust vector does not point in the same direction as the thruster. PIV images reveal that at rudder deflection angles of twenty five and thirty degrees the flow upstream of the propeller inlet has separated from the tail section and impinges at a large angle to the tail, thereby reducing both the thrust deflection angle as well as the total yaw moment acting on the vehicle.

#### **2.6.2 Introduction**

The following tasks were performed during Year 2:

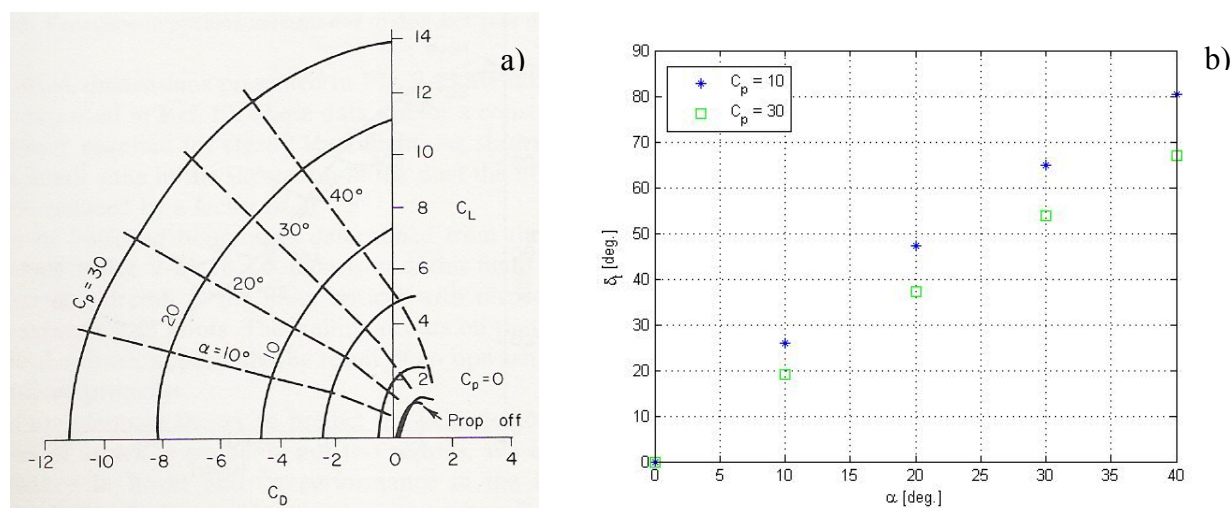
- [a] Task 3.19: Construction of experimental models and modification of flow facility mounting supports to permit reconfiguration of RPUUV control surfaces and testing at different roll, pitch and yaw positions.
- [b] Task 3.20: Experimental determination of hydrodynamic characteristics/coefficients in a 4'x4' water flume/wave tank.
- [c] Task 3.21: Test of different control surface configurations – the RPUUV was tested at rudder angles of  $0^\circ \leq \delta \leq 30^\circ$  and the resulting thrust, moment and thrust angle measured.

##### **2.6.2.1 Background**

Although the mechanical design of a vectored thruster system is more complicated than that of a fixed propeller in combination with stern and bow planes, the advantages of

using a vectored thruster system for the propulsion of small underwater vehicles such as the RPUUV, include improved maneuverability, and the fact that fewer fins, which may snag on underwater cables or other obstacles, protrude from the vehicle. This is especially important for port security applications, for example, where a vehicle must be operated in a confined, cluttered environment and possibly in strong cross currents. Presently, a limitation in the design of vectored thruster systems for small vehicles is the dearth of force and moment response data as a function of thruster angle. Many contemporary vectored thruster system models are developed under the assumptions that the propeller thrust remains constant as thruster angle changes and that the resultant thrust force is collinear with the direction of the thruster [4, 5].

The flow through a vectored thruster is complex. Firstly, an open-water ducted thruster (no upstream body) is nonlinear and cannot be modeled by simply superposing the separate effects of a duct and a propeller at different inflow angles [2, 6]. Further, as shown in Figure 2.6.1, the thrust angle of a ducted thruster does not correspond to the direction of the thruster and varies nonlinearly with duct angle. This condition will be exacerbated for the RPUUV's vectored thruster. In the presence of an upstream body, the propeller inflow is asymmetrical for non-zero thruster angles. This causes an uneven pressure distribution upstream of the rotor and thus complicates the prediction of the output force response as well as the propulsive efficiency of the thruster.

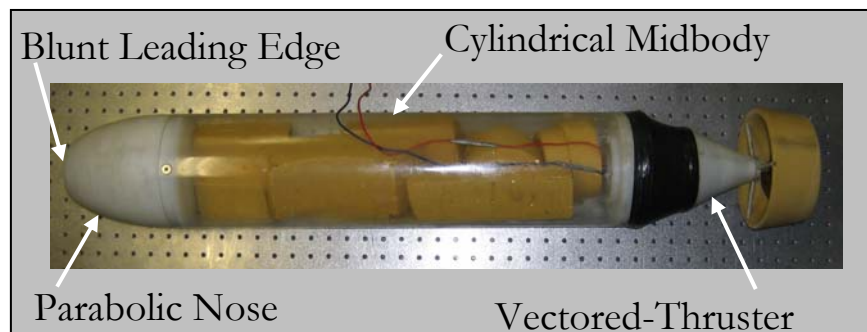


**Figure 2.6.1: a) Lift-thrust-power vs. thrust angle for a ducted rotor from [5]. Drag and lift coefficients ( $C_D$  and  $C_L$ , respectively) are plotted for several values of rotor power coefficient  $C_P$ ; contours of constant duct angle ( $\alpha$  here) are also shown. b) Direction of thrust vector  $\delta_t$  extracted from curves in a).**

Experiments were carried out on the vectored-thruster RPUUV model (Figure 2.6.2). The hull of the model has a length of  $l_v = 0.914$  m and a diameter of  $d_v = 15.24$  cm. The vectored thruster consists of a ducted propeller mounted on a gimbaled motor such that the propeller axis can trace out a cone with a half-angle of about  $40^\circ$ . A Wageningen (MARIN) 19A circular duct [3] with a chord-to-diameter ratio of  $c/D = 0.5$  is used in combination with a  $D = 13.2$  cm diameter, A-type three-bladed propeller. The propeller has a pitch/diameter ratio of 1.02 and operates with a design advance ratio of  $J = 0.31$ .



The experimental data were used to determine the hydrodynamic coefficients of the RPUUV [7, 9] and to examine the variation of both the magnitude and direction of the thrust produced as the vectored-thruster is deflected.

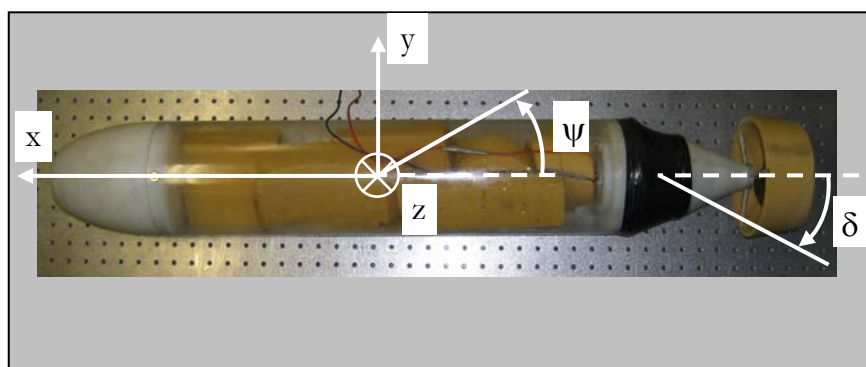


**Figure 2.6.2: The RPUUV Model.**

### 2.6.3 Experimental Setup

Experiments were conducted in the  $0.6 \times 1.22 \times 10.7 \text{ m}^3$  test section of the recirculating flume/towing tank at SeaTech. The velocity of the freestream flow was set to  $U_\infty = 0.315 \text{ m/s}$ , giving the model a Reynolds number of  $Re = 2.88 \times 10^5$ , and a Froude number of  $Fr = 0.11$  ( $Re$  and  $Fr$  are based on  $U_\infty$  and  $l_v$ ).

The forces on the RPUUV were measured for a range of thruster rudder angles  $\delta$  and vehicle yaw  $\psi$  angles (Figure 2.6.3) using a six axis force transducer (AMTI UDW3-6-100). As shown in Figure 2.6.4, the RPUUV model was suspended from a tow carriage at a depth of 0.3 m (2.0 model diameters from both the free surface and bottom of the test section). The force transducer and sting were incorporated such that as the model is rotated in yaw, the x-y axes of the model were always parallel to the x-y axes of the force transducer; the z-axes of the force transducer and model were collinear and pass through the hull midship point. An optical rotation stage mounted at the top of the sting permits  $\psi$  to be set with an accuracy of  $\pm 0.5^\circ$ . The propeller was powered by a servo-controlled DC motor. Propeller speed was set manually with a stroboscope and verified at the beginning and end of each run.



**Figure 2.6.3: Definition of angles and coordinate system**

PIV measurements of the flow were obtained using a TSI PIVCAM 10-30 camera (1024 x 1024 px<sup>2</sup> resolution, 8 bit dynamic range) with a 35 mm focal length lens set to a magnification of 1/19.2. For all experiments the flow was seeded with 11  $\mu$ m hollow glass spheres (specific gravity of 1.1 gm/cm<sup>3</sup>) and the PIV laser light sheet illumination is provided by a 532 nm Nd:YAG pulsed laser system.

As shown in Figure 2.6.4, the PIV camera was mounted above the test section. To prevent distortion effects associated with imaging through surface waves, a plexiglass viewing window was suspended just below the free surface of the test section. PIV image pairs were processed using the Hart correlation method with an interrogation region of 32 x 32 px<sup>2</sup> and 50% overlap.

Two sets of experiments were conducted in which the average thrust, drag and moments on the vehicle were measured: A) to determine the hydrodynamic coefficients of the hull and rotor duct (with the propeller removed) for  $\delta = 0^\circ$ ,  $0^\circ \leq \psi \leq 30^\circ$  and B) with  $\psi = 0^\circ$ ,  $0^\circ \leq \delta \leq 30^\circ$  and the propeller set to a constant speed. The effects of sting drag are removed from the force data by subtracting the average drag measured on the sting alone at  $\psi = 0^\circ$ . In all experiments, PIV data were simultaneously acquired with the force transducer measurements.

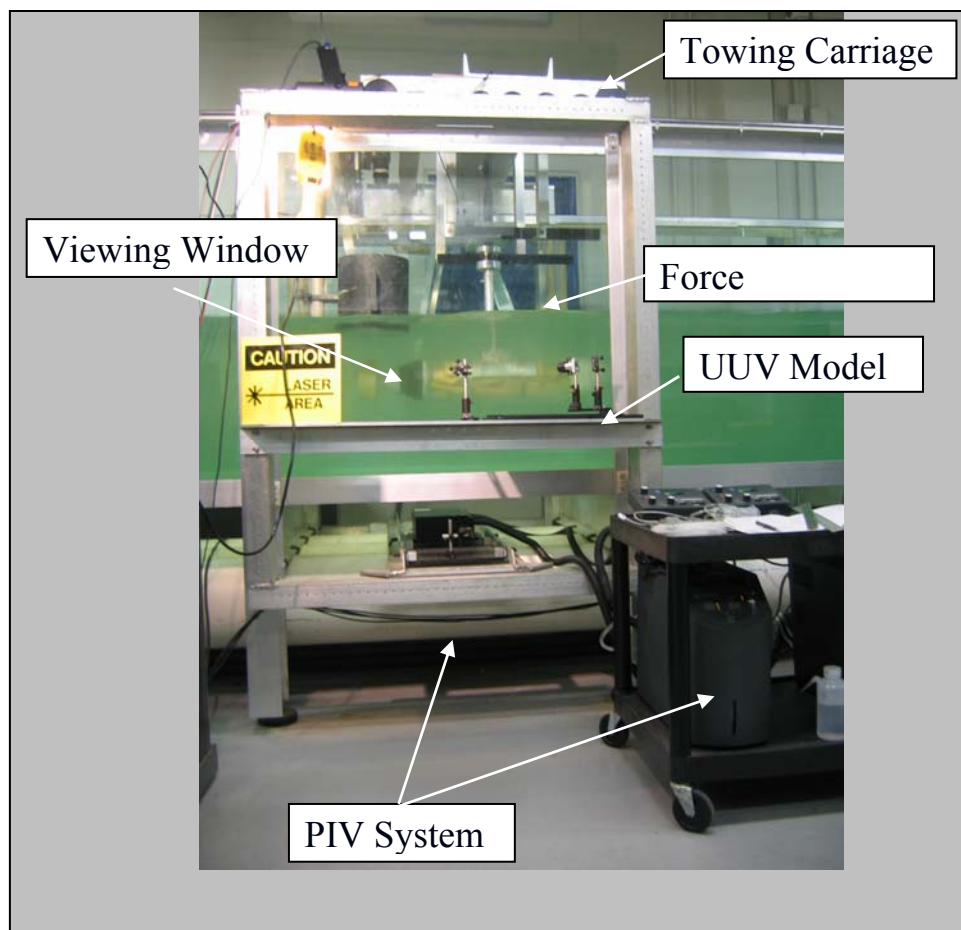


Figure 2.6.4: Experimental Setup.

## 2.6.4 Experimental Results

### 2.6.4.1 Hydrodynamic Coefficients

The drag force (total resultant force in the downstream direction), lift force (resultant force perpendicular to the freestream direction) and yawing moment are measured as  $\psi$  is varied. To obtain the drag and lift coefficients ( $C_d$  and  $C_l$ , respectively), the forces are normalized by the product of the dynamic pressure  $\frac{1}{2}\rho U_\infty^2$  and the RPUUV frontal area  $A_f = \pi d_f^2/4$ ; the yaw moment is normalized by  $\frac{1}{2}\rho U_\infty^2 A_f l_v$  to find the coefficient  $C_m$ . As shown in Figure 2.6.5 (a-c), the general trends exhibited by the  $C_d$ ,  $C_l$  and  $C_m$  coefficients are consistent with measurements made on a similar vehicle [9].

The value of  $C_d$  at  $\psi = 0^\circ$  can be predicted by theory [7] using the relation

$$C_D = C_f \left[ 1 + 60 \left( \frac{d_v}{l_v} \right)^3 + 0.0025 \left( \frac{l_v}{d_v} \right) \right], \quad (1)$$

for the drag coefficient  $C_D$  based on total wetted surface area. Here  $C_f$  is the Reynolds-number-dependent turbulent skin friction coefficient for a flat plate, for example as obtained using the I.T.T.C. line [8]. After converting  $C_D$  to  $C_d$  (drag coefficient based on  $A_f$ ), the theory predicts  $C_d \approx 0.18$ . Thus, the experimentally obtained value of  $C_d = 0.28$  at  $\psi = 0^\circ$  is only slightly higher than predicted by theory. The measured hydrodynamic coefficients, can be approximated by the following curvefits:

$$\begin{aligned} C_d &= 0.0017\psi^2 + 0.23, \\ C_l &= 0.1\psi - 0.0093, \\ C_m &= 0.011\psi + 0.037, \end{aligned}$$

which should be useful in vehicle simulations to predict the response of the vehicle to commanded inputs [7, 9, 10].

### 2.6.4.2 Variation of Thrust Output with Rudder Angle

The operational approach of many UUVs is that the propeller speed remains fixed at a constant value for the duration of a mission. Therefore, in this set of experiments the propeller speed was set to a constant value of  $n = 7.25$  Hz (435 RPM) giving an advance ratio of  $J = U/nD = 0.31$  (assuming a vehicle wake deficit of  $0.9 U_\infty$ ), the yaw angle was set to  $\psi = 0^\circ$ , and the rudder angle was varied from  $0^\circ \leq \delta \leq 30^\circ$  in increments of  $5^\circ$ .

The measured variations of thrust coefficient  $C_t$ , moment coefficient  $C_m$  and thrust angle  $\delta_t$  with rudder angle  $\delta$  are shown in Figure 2.6.6 (a-c) and the corresponding PIV results are given in Figure 2.6.7 (a-c). In contrast to the assumptions made in many numerical simulations, it can be seen that the thrust coefficient varies with rudder angle and that the direction of the thrust vector is not collinear with the duct angle. There is a sharp jump in the thrust angle when the vectored thruster has a rudder angle of between  $0^\circ$  and  $5^\circ$  suggesting that the flow transitions suddenly at small angles.

The PIV images reveal that for  $\delta = 15^\circ$ , a separated region of flow forms fore of the propeller inlet. The shear layer bounding the outer edge of this separated region is also an area high turbulent shear stress (Figure 2.6.7b,c). At  $\delta = 25^\circ$  and  $\delta = 30^\circ$  the flow behind this separation zone and fore of the propeller inlet impinges upon the articulated tail section at large angles (Figure 2.6.7c). An examination of Figure 2.6.6 (a-c) shows that at these angles the slopes of  $C_t$  and  $C_m$  as well as the corresponding thrust angle  $\delta_t$  start to decrease as a result. This suggests that the momentum of the impinging fluid at the tail reduces the side force and yaw moment on the vehicle.

### 2.6.5 Conclusions and Future Recommendations

The hydrodynamic coefficients and thrust response data are expected to be useful for predicting the open loop response of the vectored-thruster RPUUV in the field and aid with the development of a closed loop controller for the system. The magnitude of the thrust vector varies nonlinearly with rudder angle and for nonzero rudder angles the thrust vector does not point in the same direction as the thruster. PIV images reveal that at rudder deflection angles of twenty five and thirty degrees the flow upstream of the propeller inlet has separated from the tail section and impinges at a large angle to the tail. It is hypothesized that this impinging flow reduces both the thrust deflection angle as well as the total yaw moment acting on the vehicle.

The specific tasks carried out during Year 2, and presented in the report, are:

Task 3.19: Construction of experimental models and modification of flow facility mounting supports to permit reconfiguration of RPUUV control surfaces and testing at different roll, pitch and yaw positions.

Task 3.20: Experimental determination of hydrodynamic characteristics/coefficients in a 4'x4' water flume/wave tank.

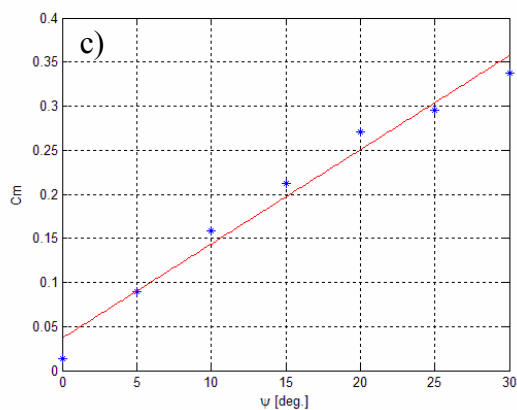
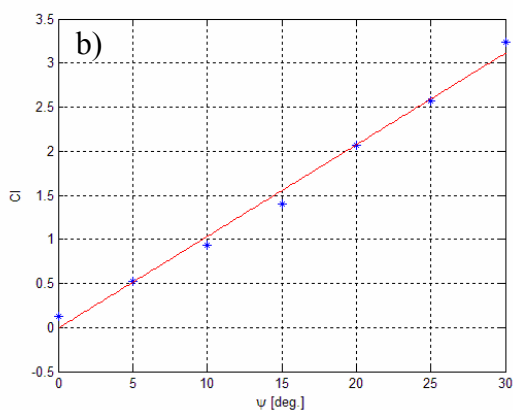
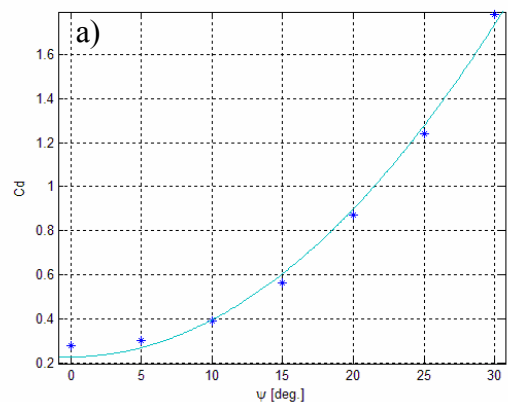
Task 3.21: Test of different control surface configurations – the RPUUV was tested at rudder angles of  $0^\circ \leq \delta \leq 30^\circ$  and the resulting thrust, moment and thrust angle measured.

Some of the key findings and future recommendations of the efforts are:

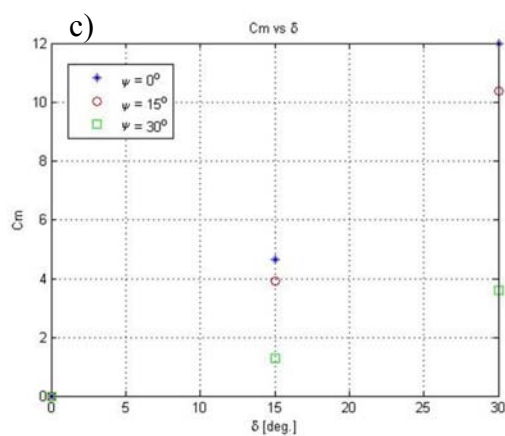
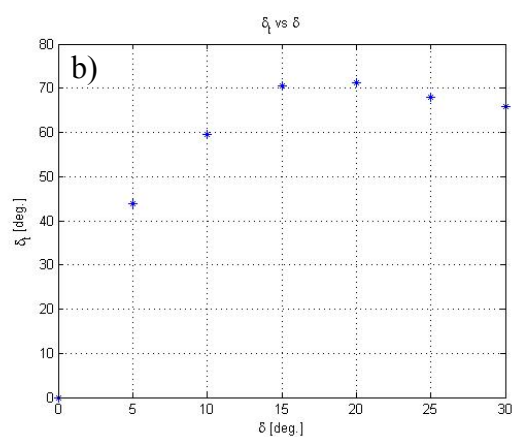
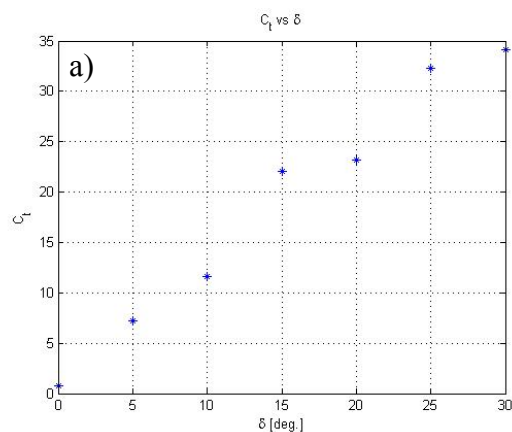
Lift, drag and moment coefficients on the RPUUV hull have been experimentally determined. Although slightly higher, the values are close to those expected from theory. Experiments demonstrate that at non-zero rudder angles the thrust produced by the vectored-thruster system is not constant and is not collinear with the symmetry axis of the thruster, as modeled in many previous numerical simulations.

The experimental data should be included in numerical simulations of the vehicle for more accurate numerical simulations of vehicle motion and for the development of an active control system.

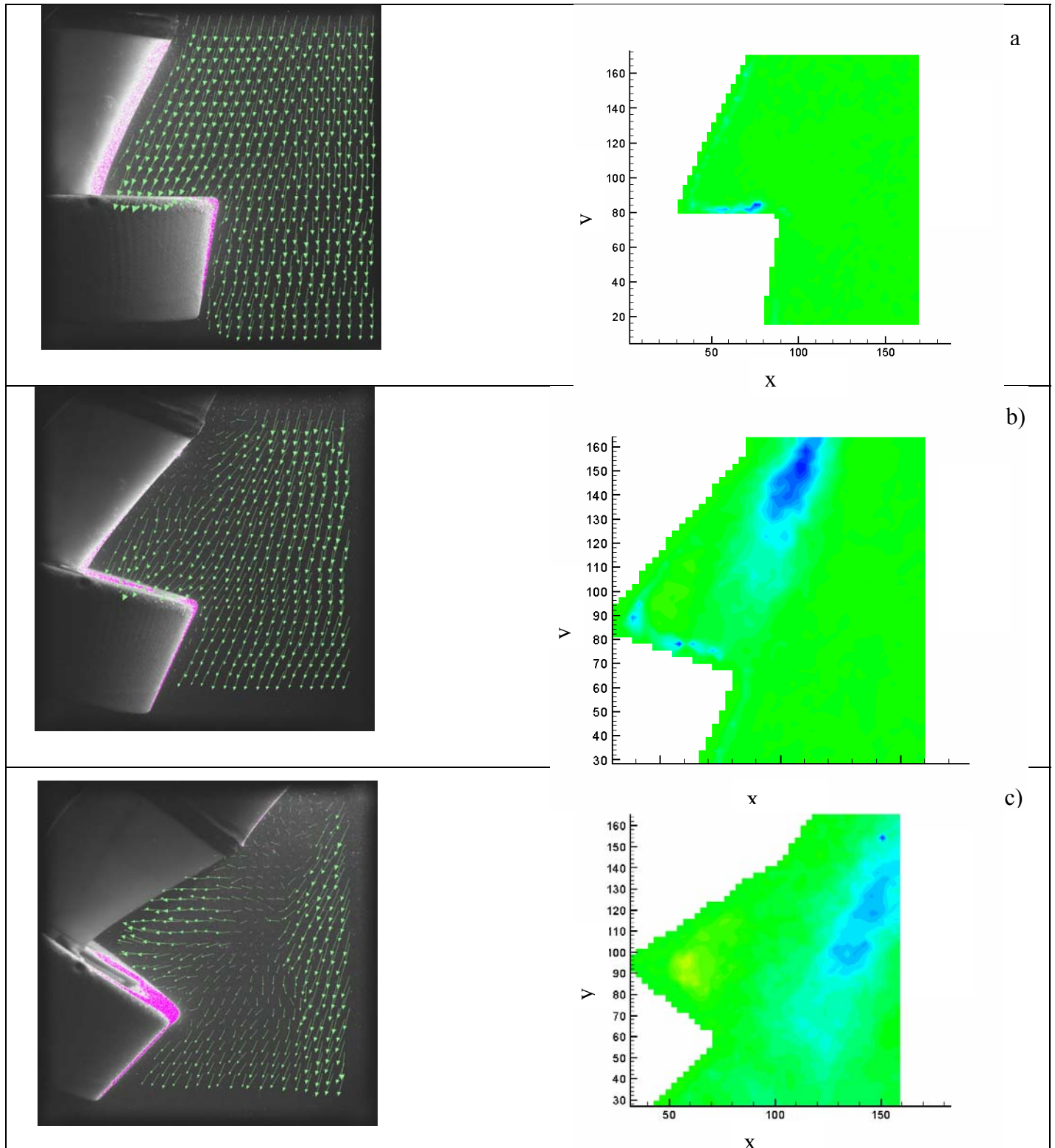
Year 3 efforts will focus on performance improvements and design modifications for the finalization of the RPUUV design; using insight gained from the three year study, alternative designs will be explored for use on future vehicles, which are specialized for port security applications.



**Figure 2.6.5: Hydrodynamic coefficients: a) drag coefficient, b) lift coefficient, and c) yawing moment coefficient.**



**Figure 2.6.6: Variation of a) thrust, b) yaw moment coefficient and c) thrust angle  $\delta_t$  as rudder angle  $\delta$  is changed.**



**Figure 2.6.7: PIV images of the flow along the articulated tail section for constant thrust case: a)  $\delta = 0^\circ$ , b)  $\delta = 15^\circ$  and c)  $\delta = 30^\circ$ . On left green arrows superimposed over image of tail section give magnitude and direction of flow, on right the separated shear layer is visualized in plots of Reynolds shear stress (blue areas).**

## REFERENCES FOR SECTION 2.6

- [1] Allen, B. Vorus, W. S. and Prestero, T. "Propulsion system enhancements on REMUS AUVs," *IEEE OCEANS 2000 MTS/IEEE Conf.*, vol. 3, 2000, pp. 1869–1873.
- [2] Bachmeyer, R. Whitcomb, L. L. and Grosenbaugh, M. A. "An accurate four quadrant nonlinear dynamical model for marine thrusters: theory and experimental validation," *IEEE J Oceanic Eng.*, vol. 25, no. 1, pp. 146-158, Jan. 2000.
- [3] Carlton, J. S. *Marine Propellers and Propulsion*. Butterworth-Heinemann, pp. 95-115, 1994.
- [4] Cavallo, E. and Michelini, R. C., "A robotic equipment for the guidance of a vectored thruster AUV," *35<sup>th</sup> Intl. Symp. Robotics*, March 2004, pp. 23-26.
- [5] Le Page, Y. G. "Hydrodynamics and Control of an Autonomous Underwater Vehicle Equipped with a Vectored Thruster," M. S. Thesis, Florida Atlantic University, Boca Raton, FL USA 2000.
- [6] McCormick, B. W. *Aerodynamics of V/STOL Flight*. Academic Press, Orlando, FL USA, pp. 249-252, 1967.
- [7] Nahon, M. "A simplified dynamics model for autonomous underwater vehicles," *IEEE. Sym. Aut. Underwater Veh. Tech.*, 1996, pp. 373–379.
- [8] Newman, J. N. *Marine Hydrodynamics*. MIT Press, Cambridge, MA USA, p. 31, 1977.
- [9] Ridley, P. Fontan, J. and Corke, P. "Submarine dynamic modeling," In *Proc. Australian Conf. Robotics & Automation*, Brisbane, QD, Australia, Dec. 2003.
- [10] Ridley, P. Fontan, J. and Corke, P. "Submarine automatic control," In *Proc. Australian Conf. Robotics & Automation*, Brisbane, QD, Australia, Dec. 2003.



## **2.7. Hydrodynamics and Dynamics Analyses of the Remotely-Piloted Unmanned Underwater Vehicle (RPUUV)**

**PI: Dr. P. Ananthakrishnan**

### **Tasks 3.22-3.25**

#### **Summary**

The Year 2 objective of the project was to carry out hydrodynamic and dynamic analyses and simulations of the remotely-piloted unmanned underwater vehicle and based on the results and findings contribute to improvements in design and performance of the vehicle. The problem formulations, solution methods, simulations, new findings and contributions are presented.

In section 2.7 of this report describes a boundary-integral algorithm based on the Green's theorem that has been developed to determine the unsteady hydrodynamic coefficients of the vehicle. Sea bottom effect are modeled based on the method of images. Results show that the hydrodynamic coefficients are only significantly affected if the vehicle is very close to the bottom. Lift and drag forces on the vehicle, appendages and fins are modeled using experimentally-determined lift and drag coefficients.

Equations governing rigid-body vehicle motion, formulated using a body-fixed frame of reference, are integrated in time using Euler's scheme to simulate vehicle dynamics. Simulations were carried out for a range of scenarios and parameter values. The vehicle, without modem and mast, are found to be dynamically robust even without any fins. The addition of an appendage such as the modem transducer induces a pitch motion which can be easily controlled using the vectored thruster. Addition of mast however induces a large unsteady pitch motion which is difficult to control either with thruster or any fixed fins. Plausible solutions to suppressing the mast-induced motions are (i) introducing a counter mast on the bottom or (ii) moving the center of gravity of the vehicle through a large distance forward; both solution are however not practical.

Dynamics of the vehicle is not affected significantly by the sea bottom even when the vehicle is very close to the bottom. The only limitation to the vehicle motion is caused by the actual bottom itself and not by the hydrodynamics aspect of the bottom.

#### **2.7.1. Introduction**

This part of the report documents the tasks related to the dynamics and hydrodynamics of the RPUUV (remotely piloted unmanned underwater vehicle) carried out during Year 2 of the project. The efforts were focused on

- development of algorithms for the determination of hydrodynamic coefficients affecting unsteady motions,
- determination of bottom effects on vehicle hydrodynamics,



- investigation of fins and vehicle configurations on vehicle motion and stability,
- simulation of vehicle motions including mast and modem transducer, and
- mission specific optimal designs.

### 2.7.1.1 Basic Vehicle Characteristics

The RPUUV consists of cylindrical middle body, hemi-spherical nose section and a conical tail section. A vectored thruster is used for forward motion as well as for maneuvering in both horizontal and vertical motions. A sketch of the vehicle, illustrating the main features of the vehicle, is given in Figure 2.7.1.1.

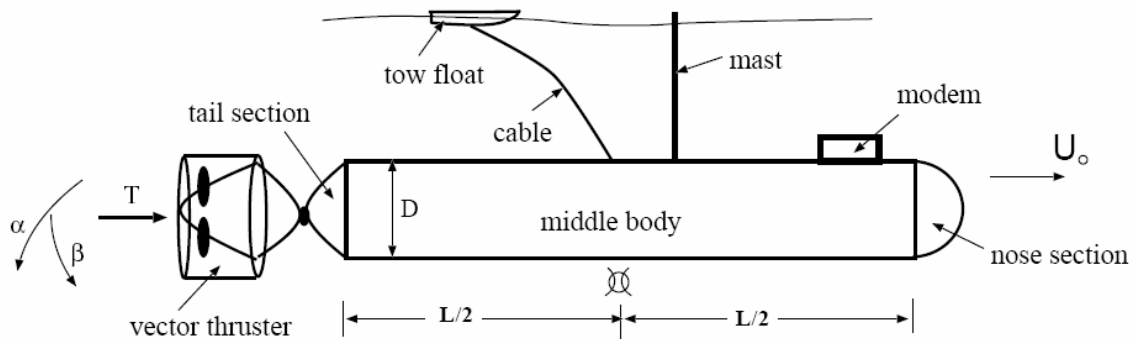


Figure 2.7.1.1 Illustrative sketch of the RPUUV

The middle body length, chosen as the characteristic length in hydrodynamic calculations is denoted as  $L$  and the vehicle diameter as  $D$ . Computations and simulations were carried out for various lengths and parameters. Results given in this report correspond to following parameter values:

- Middle-body length,  $L = 0.85$  [m]
- Diameter,  $D = 0.16$   $L = 0.136$  [m]
- Tail length =  $0.13$   $L = 0.11$  [m]
- Nose section length =  $0.08$   $L = 0.068$  [m]
- Water density,  $\rho = 1025$  [kg/m<sup>3</sup>]
- Acceleration of gravity,  $g = 9.8$  [m/s<sup>2</sup>]
- Vehicle volume =  $0.0219$   $L^3 = 0.01345$  [m<sup>3</sup>]
- Vehicle mass (for neutral buoyancy),  $m = 13.78$  [kg] = 30 [lbf]
- Wetted surface area =  $0.578$   $L^2 = 0.42$  [m<sup>2</sup>]
- Projected area normal to x axis,  $A_p = 0.02$   $L^2 = 0.0145$  [m<sup>2</sup>]
- Mast height = 1 to 1.5 [m]
- Mast diameter = 1/8 inch = 0.32 [cm]
- Modem height = 2.5 inch = 0.063 [m]
- Modem diameter = 2.5 inch = 0.063 [m]

#### **2.7.1.2 Vehicle Motion**

The equations governing the vehicle motion are formulated using the body-fixed coordinates  $xyz$  with origin fixed at the intersection of the vehicle axis and mid-section. The equations are solved using finite-difference time integration to simulate vehicle motions in both horizontal and vertical planes. The formulation is presented in Section 2.7.2

#### **2.7.1.3 Hydrodynamics of the RPUUV**

A boundary-integral method based on the Green's theorem is used to determine hydrodynamic coefficients for the unsteady vehicle motion, both in infinite fluid and including effect of bottom boundary. The viscous drag force is determined using experimentally-obtained drag coefficient. The lift and drag forces on the fins, mast, modem and other appendages are determined empirically based on the lift and drag coefficients taken from literature. These methodologies are described in Section 2.7.3 of this report.

#### **2.7.1.4 Motion Simulations**

Simulation of vehicle motion both in horizontal and vertical planes are carried out to determine effects of various parameters and sensors on vehicle performance. The case studies carried out include, (i) vertical plane motion including forces on mast and modem, (ii) horizontal and vertical plane motions for various thrust angles, (iii) vehicle motion including fins and (iv) horizontal plane motion close to the bottom. The simulations and findings are elaborated in Section 2.7.4 of the report.

#### **2.7.1.5 Contributions of the Project**

Finally, in Section 2.7.5 of the report, the contributions of the project to design and for improved vehicle performance are summarized. Ongoing Year 3 efforts are outlined.

### 2.7.2. Formulation of Vehicle Motion

The equations of rigid-body motion are formulated using a body fixed frame of reference  $oxyz$  as shown in Figure 2.7.2.1. The  $x$  axis, which is the axis of symmetry, is positive forward. The  $y$  axis is from port to starboard and the  $z$  axis is downward. The  $x$ ,  $y$  and  $z$  components of translational velocity are denoted as  $u$ ,  $v$  and  $w$  and the rotational velocity as  $p$ ,  $q$  and  $r$ . The steady component of the forward velocity is denoted as  $U_0$ . Propeller thrust is denoted as  $T$ .

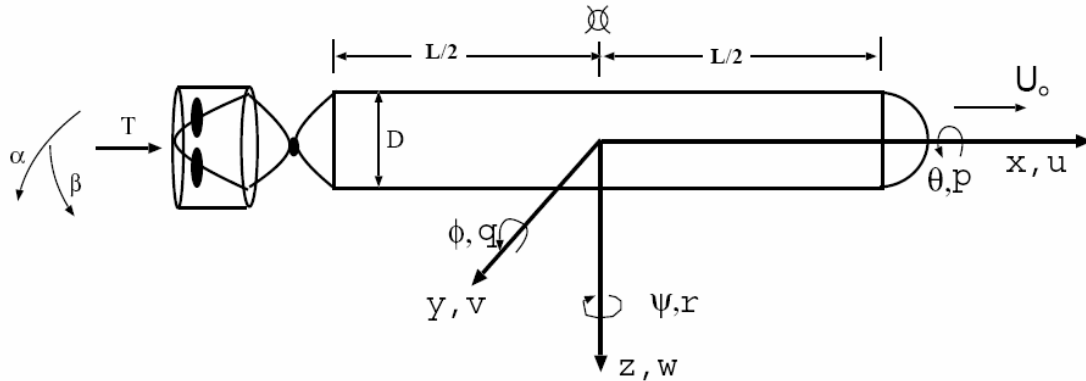


Figure 2.7.2.1 Body-fixed coordinates and notations used in the RPUUV dynamics formulation

#### 2.7.2.1 Six DOF Rigid Body Equations of Motion

The 6DOF equations of rigid body motion with respect to body-fixed coordinates are given by [1], [7]

##### 6DOF-SURGE

$$m[\dot{u} - vr + wq - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] = \mathcal{X}$$

##### 6DOF-SWAY

$$m[\dot{v} - wp + ur - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})] = \mathcal{Y}$$

##### 6DOF-HEAVE

$$m[\dot{w} - uq + vp - z_G(p^2 + q^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})] = \mathcal{Z}$$

### 6DOF-ROLL

$$I_x \dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} + m[y_G(\dot{w} - uq + vp) - z_G(\dot{v} - wp + ur)] = \mathcal{K}$$

### 6DOF-PITCH

$$I_y \dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} + m[z_G(\dot{u} - vr + wq) - x_G(\dot{w} - uq + vp)] = \mathcal{M}$$

### 6DOF-YAW

$$I_z \dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} + m[x_G(\dot{v} - wp + ur) - y_G(\dot{u} - vr + wq)] = \mathcal{N}$$

In the above equations  $m$  denotes the vehicle mass and ( $I_x$ ,  $I_y$  and  $I_z$ ) the mass moments of inertia about  $x$ ,  $y$  and  $z$  axis, respectively. The coordinates of the center of gravity are denoted as ( $x_G$ ,  $y_G$ ,  $z_G$ ). The  $x$ ,  $y$  and  $z$  components of the resultant external force are denoted as  $X$ ,  $Y$  and  $Z$ , respectively. The components of the moment of the external force about  $x$ ,  $y$  and  $z$  axes are denoted as  $K$ ,  $M$  and  $N$ , respectively. The over-dot represents time derivative.

#### **2.7.2.2 Horizontal Plane Three DOF Equations of Motion**

The primary modes affecting the motion on the horizontal plane are surge, sway and yaw. Setting other motions to be zero, we can obtain the following equations for the horizontal plane motion:

$$m(\dot{u} - vr - x_G r^2 - y_G \dot{r}) = \mathcal{X},$$

$$m(\dot{v} + ur + x_G \dot{r} - y_G r^2) = \mathcal{Y},$$

$$I_z \dot{r} + m[x_G(\dot{v} + ur) - y_G(\dot{u} - vr)] = \mathcal{N}$$

In the case of the plane motion, note that  $r = d\psi/dt$  where  $\psi$  denotes the Euler angle of displacement about the  $z$  axis (ie. heading angle).

### 2.7.2.3 Vertical Plane Three DOF Equations of Motion

The equations governing surge, heave and pitch motions are given by

$$\begin{aligned} m(\dot{u} + wq - x_G \dot{q}^2 + z_G \dot{q}) &= \mathcal{X}, \\ m(\dot{w} - uq - z_G \dot{q}^2 - x_G \dot{q}) &= \mathcal{Z}, \\ I_y \dot{q} - m[x_G(\dot{w} - uq) - z_G(\dot{u} + wq)] &= \mathcal{M} \end{aligned}$$

Note that the righting moment associated with the meta-centric height is included  $M$ .

### 2.7.2.4 Method of Analysis

The modeling of external force and moment, which appear on the right-hand side of above equations of motion, is explained in Section 2.7.3 of the report. With initial values specified, the equations governing the body motion subject to external force and moment are time-integrated using the Euler's scheme to advance the solution in time. Upon determining velocity components in the body-fixed frame, the velocity components in the earth-fixed frame are obtained by appropriate coordinate transformation. The earth-fixed velocity components are then integrated to track vehicle trajectory and orientation. The vehicle motions are thus simulated.

### 2.7.3. Determination of Hydrodynamic Forces and Moments

The hydrodynamic forces acting on the main hull due to unsteady motions are determined using the added-mass theory. The added-mass and -moment coefficients are computed by solving the Green's theorem using a boundary-integral method. The viscous drag force is determined using drag coefficients obtained from experiments. The forces on fins, mast, modem and other appendages are determined using drag and lift coefficients given in the literature.

#### 2.7.3.1 Unsteady Flow Hydrodynamics

The unsteady hydrodynamics related coefficients can be determined using potential flow theory, as effect of viscosity on such coefficients is not significant. The equations governing the potential flow are given by [4]

$$\nabla^2 \Phi = 0$$

where  $\Phi$  denotes the velocity potential:  $\mathbf{u} = \text{grad } \Phi$ . The above Laplace equation is governed by the following boundary conditions:

$$\frac{\partial \Phi}{\partial n} = V_n \quad \text{on the body surface } S_B$$

and

$$\Phi \rightarrow 0 \quad \text{at } \infty$$

where  $V_n$  denotes the normal component of the body velocity due to rigid body motion:

$$V_n = \vec{U} \cdot \hat{n} + \vec{r} \times \vec{\Omega} \cdot \hat{n}$$

Here  $\mathbf{n}$  denotes the normal (inward) velocity on the body surface,  $\mathbf{U}$  the translational velocity, and  $\mathbf{r} \times \mathbf{\Omega}$  the transverse velocity due to rotation of the body. By way of vector identity and index notation [4], the above body boundary condition can be written as

$$\begin{aligned} V_n &= \vec{U} \cdot \hat{n} + \vec{\Omega} \cdot (\vec{r} \times \hat{n}) \\ &= U_i n_i \end{aligned}$$

where

$$\begin{aligned} U_i &= (u, v, w) \quad \text{for } i = 1, 2, 3 \quad (\text{translational modes}) \\ &= (p, q, r) \quad \text{for } i = 4, 5, 6 \quad (\text{rotational modes}) \end{aligned}$$

and

$$\begin{aligned} n_i &= (n_x, n_y, n_z) \quad \text{for } i = 1, 2, 3 \quad (\text{components of normal vector}) \\ &= (\vec{r} \times \hat{n})_1, (\vec{r} \times \hat{n})_2, (\vec{r} \times \hat{n})_3 \quad \text{for } i = 4, 5, 6 \quad (\text{components of the moment of normal vector}) \end{aligned}$$

The above body-boundary condition suggests the following model decomposition (Kirchoff's decomposition) of the total velocity potential:

$$\Phi = U_i \phi_i, \quad (\text{where } i = 1, 2, 3, 4, 5, 6)$$

where  $\phi$  is referred to as the unit potential corresponding to the  $i$ -th mode of body motion. Substituting the above decomposition in the equations governing  $\Phi$ , one can obtain the following equations for the unit potentials:

$$\nabla^2 \phi = 0$$

$$\phi_i \rightarrow 0, \quad \text{at } \infty$$

$$\frac{\partial \phi_i}{\partial n} = n_i, \quad \text{on the body surface } S_B$$

Upon solving the above equations for unit potential, one can define and determine added-mass (and moment) tensor coefficients as [4]

$$\mu_{ij} = \rho \int_{S_B} \phi_i n_j dS_B$$

### 2.7.3.2 Determination of Unsteady Hydrodynamic Forces and Moments

Based on the principle of balance of linear momentum, and in terms of added-mass coefficients, one can establish the following relations for the hydrodynamic force and moment acting on the body (see Newman [4] for details):

$$F_j = -\frac{d}{dt} U_i \mu_{ji} - \epsilon_{jkl} U_i U_{k+3} \mu_{li}, \quad \text{where } j,k,l=1,2,3; \text{ and } i=1,2,\dots,6$$

$$M_j = -\frac{d}{dt} U_i \mu_{j+3,i} - \epsilon_{jkl} U_i U_{k+3} \mu_{l+3,i} - \epsilon_{jkl} U_i U_k \mu_{li}, \quad \text{where } j,k,l=1,2,3; \text{ and } i=1,2,\dots,6$$

For compactness, the index notation is used in the above representations. The notation  $\epsilon_{jkl}$  is called the Kronecker delta and it stands for

$$\begin{aligned} \epsilon_{jkl} &= +1, \quad (\text{for } jkl = 123, 231, 312) \\ &= -1, \quad (\text{for } jkl = 132, 213, 321) \\ &= 0, \quad (\text{for all other permutations of } jkl) \end{aligned}$$

Note that  $F_1, F_2, F_3$  correspond to  $x, y$  and  $z$  components of the hydrodynamic force and  $M_1, M_2, M_3$  to the  $x, y$  and  $z$  components of the moment of the hydrodynamic force. Our task now is to solve for the unit potentials which are required for the evaluation of the added-mass and -moment coefficients and therefore the hydrodynamic forces and moments. The task is accomplished using a boundary-integral algorithm based on the Green's theorem.

### 2.7.3.3 Green's Theorem and Boundary-Integral Method

The Green's theorem governing body motion in infinite fluid is given by [4]

$$2\pi\phi_i(P) + \int_{S_B, P \neq Q} \phi_i(Q) \frac{\partial}{\partial n_Q} \frac{1}{r} dS_B(Q) = \int_{S_B, P \neq Q} \frac{1}{r} \frac{\partial \phi_i}{\partial n_Q} dS_B(Q)$$

where the Green's function correspond to potential due to a point source:

$$G \equiv \frac{1}{r}$$

with  $r$  being the distance between the source point  $Q$  and the field point  $P$ , as illustrated in Figure 2.7.3.1. Note that in the above equation, the field point  $P$  is on the body surface.

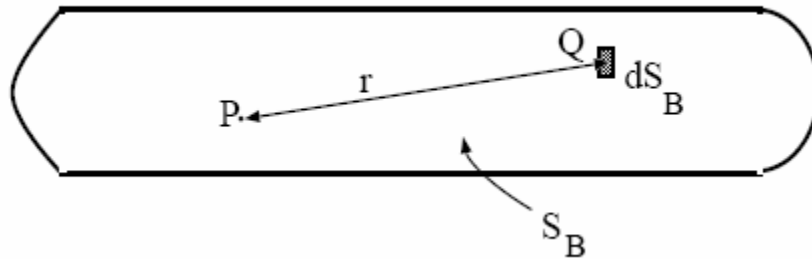


Figure 2.7.3.1 Green's theorem for body motion in infinite fluid.

The above integral equation is to be solved for the unit potentials  $\phi_i$  with  $i=1,2,\dots,6$ . Upon discretization of the body surface, as illustrated in Figure 2.7.3.2, the integral equation can be converted to a matrix equation given by [5]

$$A_{kl}\phi_{i,l} = b_k$$

where  $A_{kl}$  denotes the coefficient matrix corresponding to the left side of the integral equation and  $b_k$  the right side of the integral equation. Note that the right side is known, because

$$\frac{\partial \phi_i}{\partial n_Q} = n_i \quad (\text{no-flux condition})$$

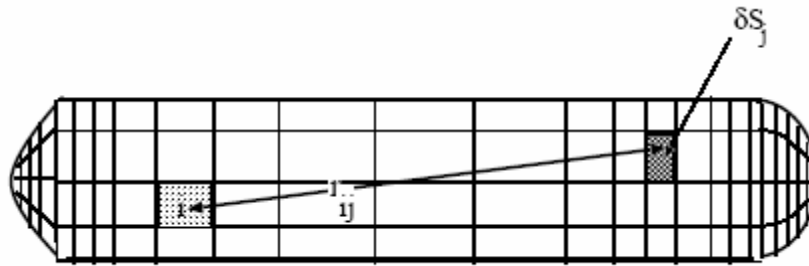


Figure 2.7.3.2 Discretization of the body surface

The matrix equation can be solved using, for example, Gauss-Jordan algorithm for the unit potential  $\phi_i$  corresponding to the  $i$ -th mode of motion. Repeating six times, the unit potentials corresponding to all six modes of rigid-body motion can be determined. By numerical integration of

$$\mu_{ij} = \rho \int_{S_B} \phi_i n_j dS_B$$

the added-mass and -moment coefficients  $\mu_{ij}$  for all six modes of rigid body motion are then obtained.



### 2.7.3.4 Effect of Sea Bottom on Hydrodynamic Coefficients

Hydrodynamic formulation and analysis for vehicle motion over sea bottom, as illustrated in Figure 2.7.3.3, require some modifications. The unit potentials  $\phi_i$  now has to satisfy an additional boundary condition:

$$\frac{\partial \phi_i}{\partial n} = 0, \quad (\text{on the bottom})$$

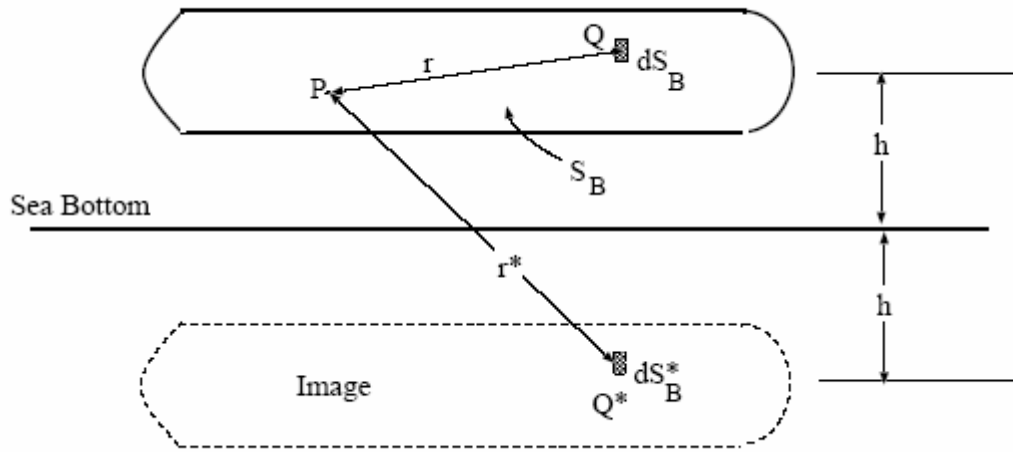


Figure 2.7.3.3 RPUUV motion above sea bottom: modeling using method of images.

The bottom no-flux condition can be easily implemented using the method of images. In other words, the vehicle moving over the bottom is equivalent to two RPUUVs moving in infinite fluid as shown in Figure 2.7.3.3. However, the boundary-integral analysis need not involve integrals over both the RPUUV and its image. Using the Green's function

$$G = \frac{1}{r} + \frac{1}{r^*}$$

where  $1/r$  denotes the distance between field point and source point and  $1/r^*$  the distance between the source point and the source's image point, the Green's theorem for the present case becomes

$$2\pi\phi_i(P) + \int_{S_B, P \neq Q} \phi_i(Q) \frac{\partial}{\partial n_Q} \left( \frac{1}{r} + \frac{1}{r^*} \right) dS_B(Q) = \int_{S_B, P \neq Q} \left( \frac{1}{r} + \frac{1}{r^*} \right) \frac{\partial \phi_i}{\partial n_Q} dS_B(Q)$$

As in case of RPUUV motion in infinite fluid, the above integral equation when discretized yields a matrix equation which can be solved for the unit velocity potentials. Note that the solution is dependent on the altitude  $h$  and therefore the unit potentials have to be computed for various altitude distances of interest.

### 2.7.3.5 Viscous Drag Force

The viscous drag force  $F_D$  on the hull and appendages are determined in terms of drag coefficient as

$$F_D = C_D \frac{1}{2} \rho u |u| A_p$$

$U$  here denotes the velocity of the body in the direction of  $F_D$  and  $A_p$  the projected area of the body normal to the direction of vehicle motion. Experimentally obtained values of  $C_D$  [3] [6] were used in the calculations.

For rotational modes, the cross drag force is estimated based on strip theory. At each strip, the drag force is determined based on the transverse velocity of the strip due to body rotation. The total force is taken as the sum of all the strip forces. The moment of the drag force is taken to be the sum of moment of force on each strip.

### 2.7.3.6 Fin Force

The lift  $F_L$  drag  $F_D$  forces acting on a fin are computed based on respective coefficients:

$$F_L = C_L \frac{1}{2} \rho U^2 A_f; \quad F_D = C_D \frac{1}{2} \rho U^2 A_f,$$

where  $C_L$  and  $C_D$  denote lift and drag coefficients, respectively,  $U$  the resultant velocity of the fin and  $A_f$  the fin plan-form area (= span x chord). Experimentally obtained values of  $C_L$  and  $C_D$  [2] [3] were used in the calculations.

### 2.7.3.7 Other External Forces and Moments

The propeller thrust  $T$  and the angles of vectored thruster about horizontal and vertical planes are specified in the simulations. Modeling of the vectored thruster, specifically propeller thrust .vs. rpm relation, is a part of an accompanying project by PI von Ellenrieder. The righting moment due to pitch motion, for small angles, is given by

$$M = \Delta \cdot GM \cdot \sin \phi \approx \Delta \cdot GM \cdot \phi$$

where  $\Delta$  denotes the weight of the vehicle,  $GM$  the metacentric height which is equal to the vertical distance between the centers of buoyancy and gravity and  $\phi$  the pitch angular displacement.

The calculated values of hydrodynamic forces and moments, as described above, are given in Table 2.7.3.1. The computations were carried out in terms of non-dimensional variables, non-dimensionalized with respect to water density  $\rho$ , acceleration of gravity  $g$  and mid-body length  $L$ . The dimensional values given in the table correspond to  $\rho = 1025 \text{ [kg/m}^3\text{]}$ ,  $g = 9.8 \text{ [m/s}^2\text{]}$  and  $L = 0.85 \text{ [m]}$ . The indices 1,2 and 3 in added-mass coefficients correspond to linear motions along  $x$ ,  $y$  and  $z$  directions respectively, and 4,5 and 6 to rotational motions about  $x$ ,  $y$  and  $z$ , respectively. Simpler expressions, deduced from the complete formulas for the specific vehicle, are also given in the table.

Table 2.7.3.1 Principal Geometry and Hydrodynamics Related Quantities.

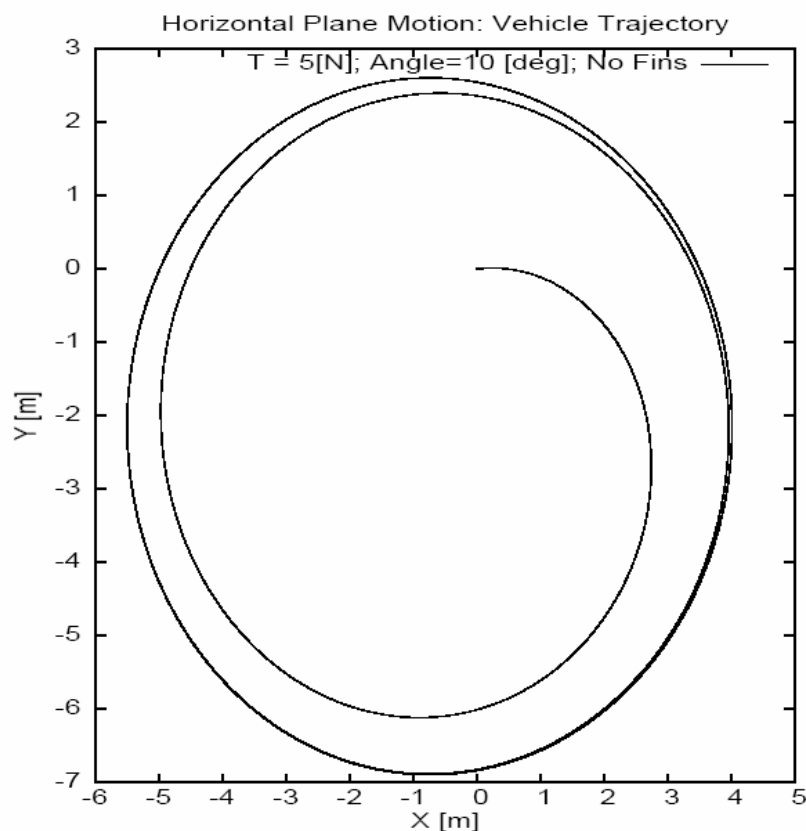
Item	Value	Remarks
Vehicle mass, $m$	13.78 [kg]	neutrally buoyant
Drag force at $U_O = 1 \text{ [m/s]}$	1.5 [N]	$C_D = 0.2$ $D = 1.5 U^2$
Modem drag at $U_O = 1 \text{ [m/s]}$	2.07 [N]	$C_D = 1.0$ $D = 2.07 U^2$
Modem drag moment at $U_O = 1 \text{ [m/s]}$	0.207 [N m]	$C_D = 1.0$ $M = 0.207 U^2$
Mast drag at $U_O = 1 \text{ [m/s]}$	2.44 [N]	$C_D = 1.0$ $D = 2.44 U^2$
Mast drag moment at $U_O = 1 \text{ [m/s]}$	2.205 [N m]	$C_D = 1.0$ $M = 2.205 U^2$
Added mass $\mu_{11}$	0.48 [kg]	in infinite fluid
Added mass $\mu_{22} = \mu_{33}$	11.6 [kg]	in infinite fluid
Added mass $\mu_{26} = \mu_{53}$	0.02 [kg-m]	in infinite fluid
Added mass $\mu_{55} = \mu_{66}$	0.62 [kg-m <sup>2</sup> ]	in infinite fluid

### 2.7.4. Simulation of Vehicle Motions: Discussions and Findings

In this section of the report, we present and discuss results of numerous simulations of RPUUV motion carried out during Year 2 of the project. In order not to overwhelm the reader with plots and graphs, we present only the key results relevant to design and performance of the vehicle. All the simulations were carried out from  $t = 0$  to  $t = 100$  [s]. The thruster was ramp started from  $t = 0$  [s] so as to reach specified thrust at  $t = 30$  [s].

#### 2.7.4.1 Horizontal Plane Motion With and Without Aft Fins

First, we present results for horizontal plane motion of the RPUUV without aft fins. The vehicle trajectory corresponding to thrust of 5 [N] thrust angle  $\alpha$  of 10 [deg] is shown in Figure 2.7.4.1 As can be observed, the vehicle executes a circle of radius 5 [m].



**Figure 2.7.4.1** Trajectory of RPUUV without fins:  $T = 5$  [N], thruster angle  $\alpha = 10$  [deg] and  $x_G = +0.035$  [m]

Next, trajectory of the vehicle with aft fins and corresponding to  $T = 5$  [N] and thrust angle  $\alpha = 10$  [deg] presented (Figure 2.7.4.2). Both chord and span length of each fin is set to be 0.05 [m] and the slope of the lift-coefficient curve  $dC_L/d\delta$  as 9.0 [per radian]. The presence of fins makes the vehicle only turn gently and not execute motions in a tight circle. Increase of thruster angle to 40 [deg] enables the vehicle to execute a circle of radius 20 [m], as shown in Figure 2.7.4.3.

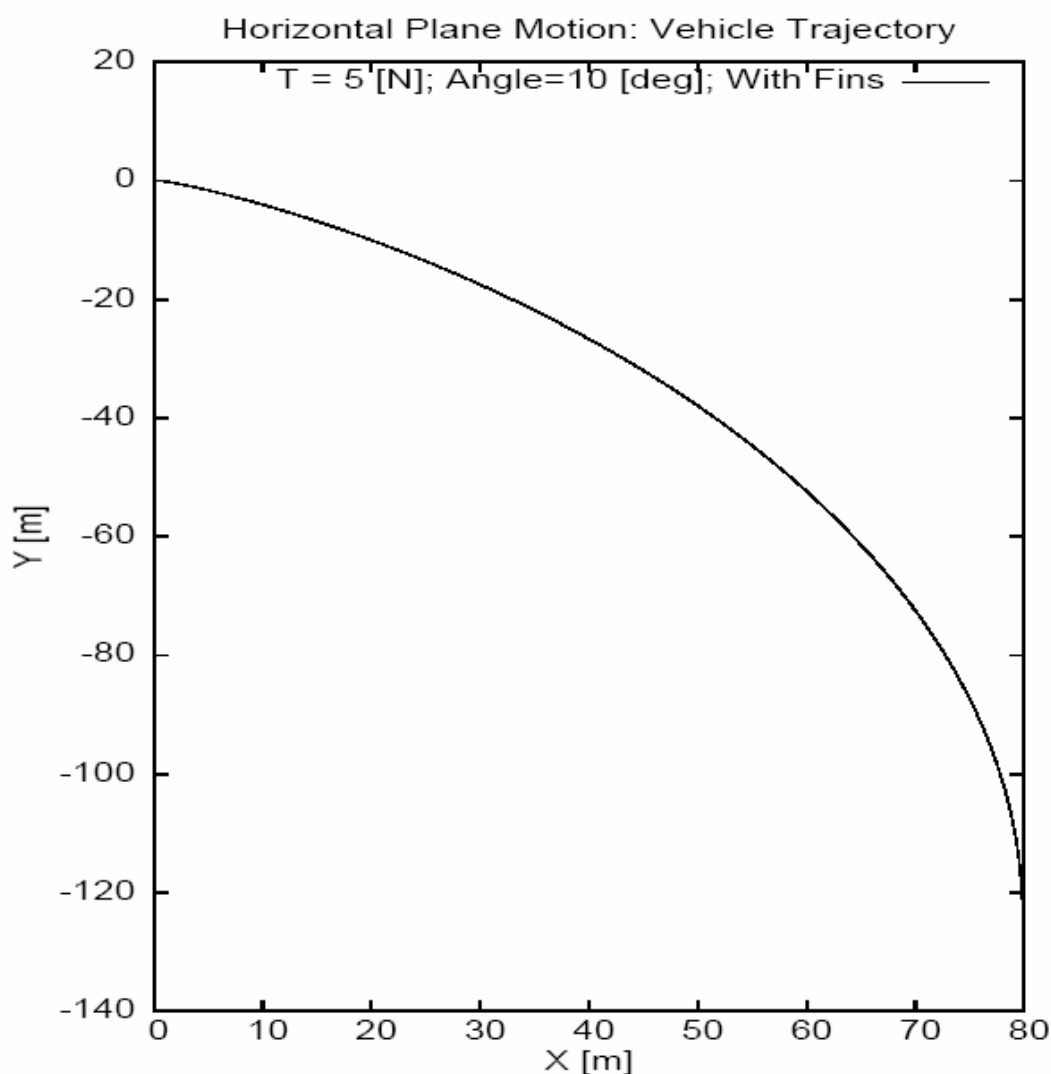
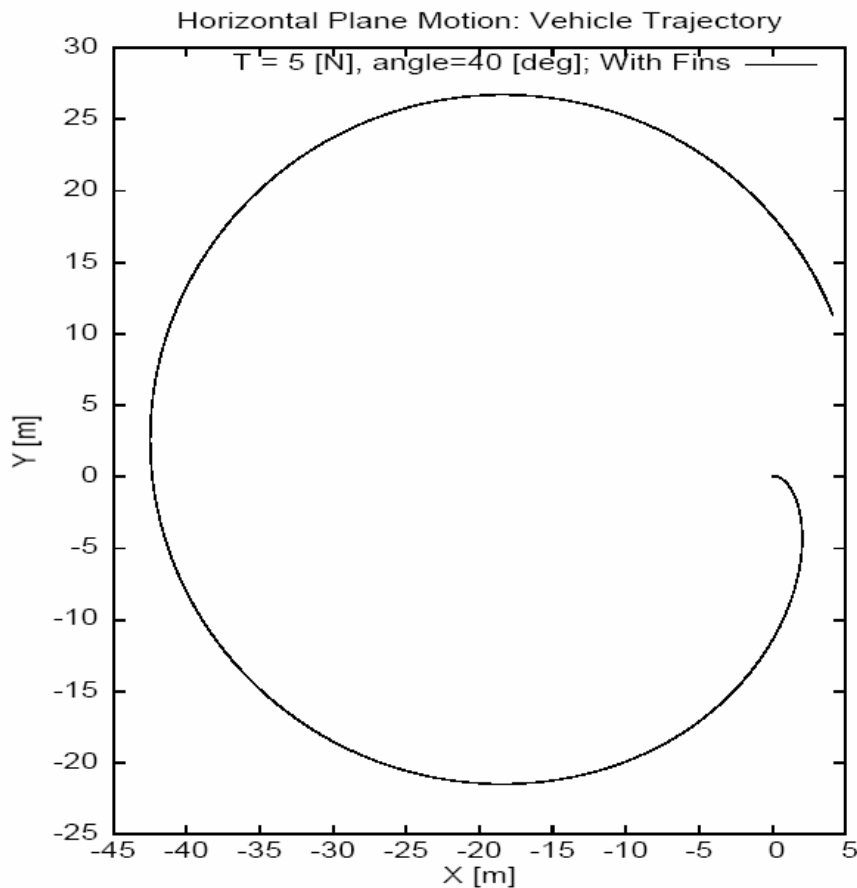


Figure 2.7.4.2 Trajectory of RPUUV with fins:  $T = 5$  [N], thruster angle  $\alpha = 10$  [deg] and  $x_G = +0.035$  [m], chord = span = 0.05 [m]; fin distance = 0.4 [m] aft.



**Figure 2.7.4.3** Trajectory of RPUUV with fins:  $T = 5$  [N], thruster angle  $\alpha = 40$  [deg] and  $x_G = +0.035$  [m], chord = span = 0.05 [m]; fin distance = 0.4 [m] aft.

Even though not shown in the figure, simulations were carried out to test the directional stability of the vehicle. All above runs were repeated with introduction of a perturbation in the form of a spike in sway velocity. In all runs, the perturbation vanished rapidly and there was not any significant change to vehicle motion subsequently.

#### **2.7.4.2 Vertical Plane Motion With and Without Aft Fins**

Next, simulated results corresponding to vertical plane motion (surge, heave and pitch) of the RPUUV with and without aft fixed fins are presented. All the simulations were carried out from  $t = 0$  to  $t = 100$  [s]. The thruster was ramp started from  $t = 0$  [s] so as to reach specified thrust at  $t = 30$  [s]. Representative results related to design and performance is discussed.

Result in the form of vehicle trajectory corresponding to thrust  $T = 5$  [N], thrust angle =  $10$  [deg] (positive upwards) and without fins is shown in Figure 2.7.4.4. The vehicle is stable and executes a dive that spans  $180$  [m] forward and  $100$  [m] downward over a duration of  $100$  [s].

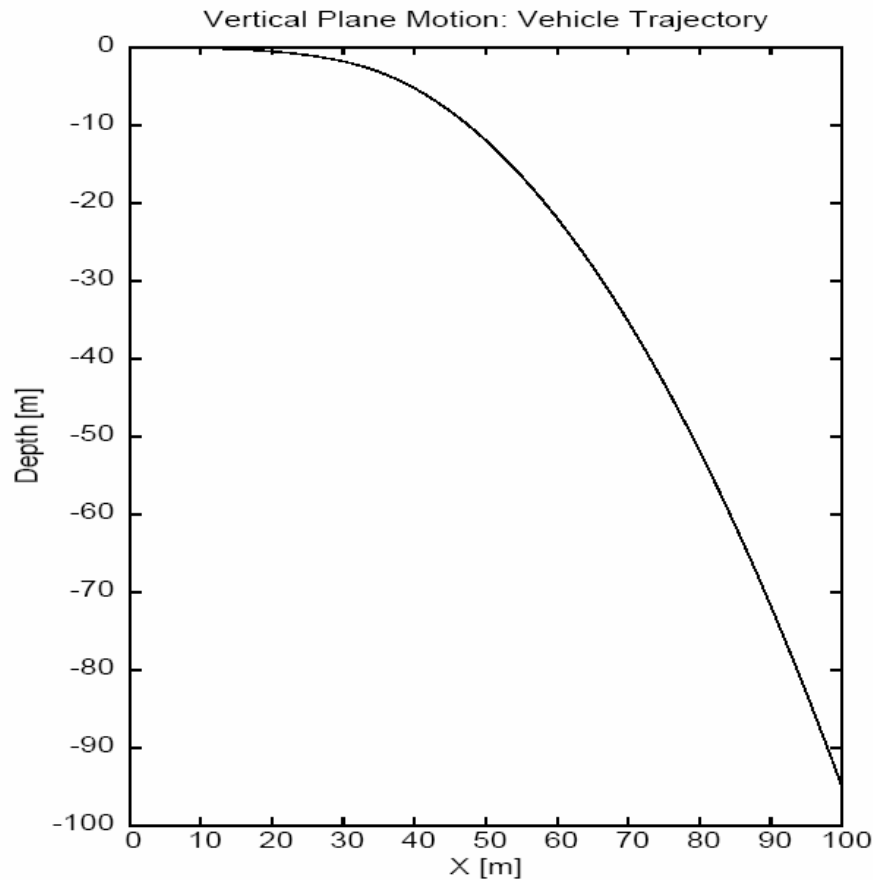


Figure 2.7.4.4 Trajectory of RPUUV (in the vertical plane) without fins:  $T = 5$  [N], thruster angle  $\alpha = 10$  [deg] and  $x_G = +0.035$  [m].

Result corresponding to same above case but with aft fins (fixed stern-planes) of chord =  $0.05$  [m], span =  $0.05$  [m], fin distance from mid-section =  $0.4$  [m] and lift-coefficient per unit angle of attack =  $9$  [per radian] is given in Figure 2.7.4.5. The presence of fins causes the body to rise by  $7$  [m] over a horizontal span of  $150$  [m] over a period of  $100$  [s].

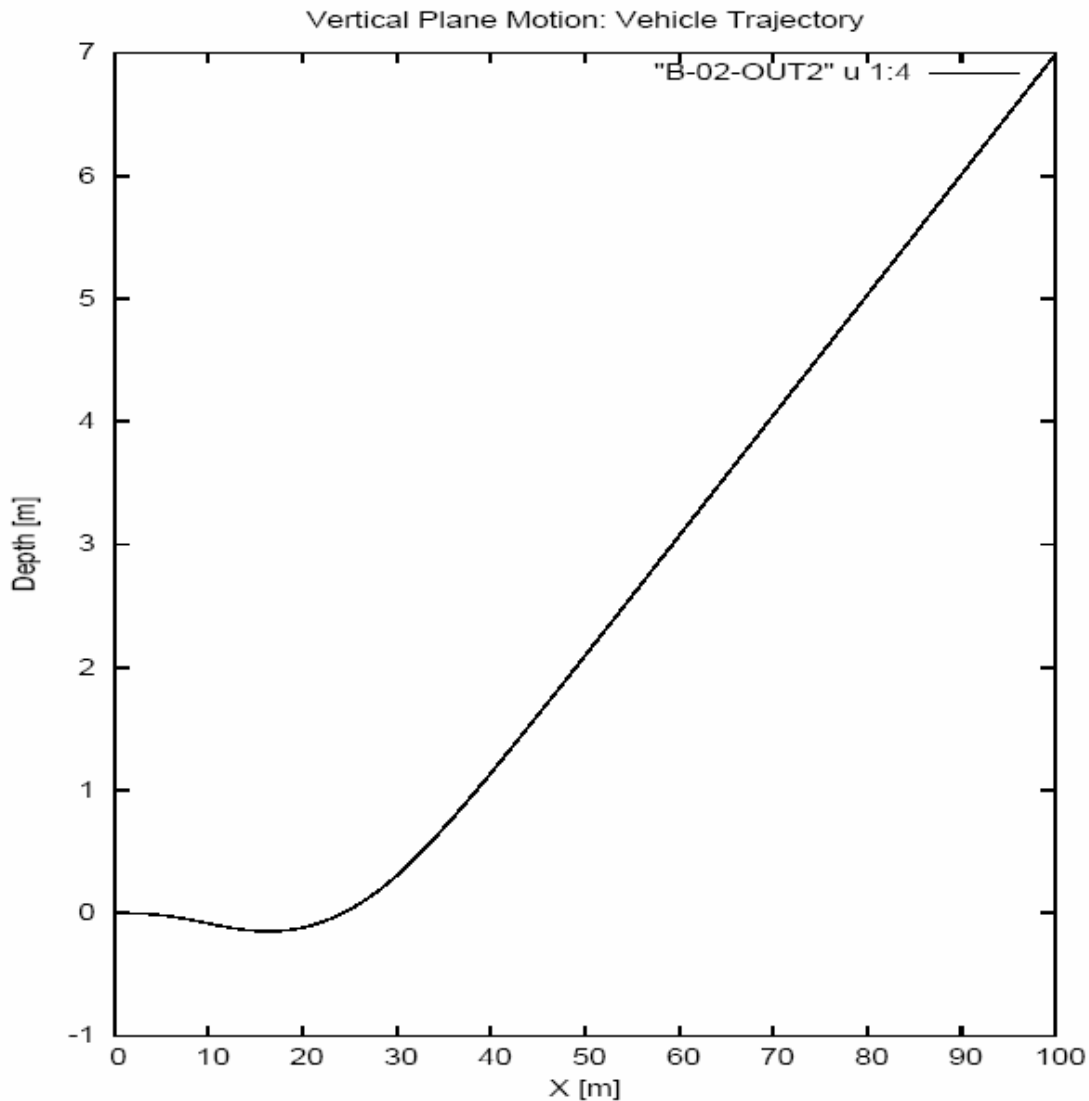


Figure 2.7.4.5 Trajectory of RPUUV (in the vertical plane) with fins:  $T = 5$  [N], thruster angle  $\alpha = 10$  [deg] and  $x_G = +0.035$  [m], chord = span = 0.05 [m]; fin distance = 0.4 [m] aft.

Though not presented in this report, simulations showed that the vehicle is dynamically stable in that any perturbation introduced to vehicle motion vanishes rapidly in time.

Results obtained for large vector thrust angles and with aft fins are given in Figure 2.7.4.6. The vehicle can dive or surface, even with aft fins, by appropriately setting vector thruster angles in the vertical plane.



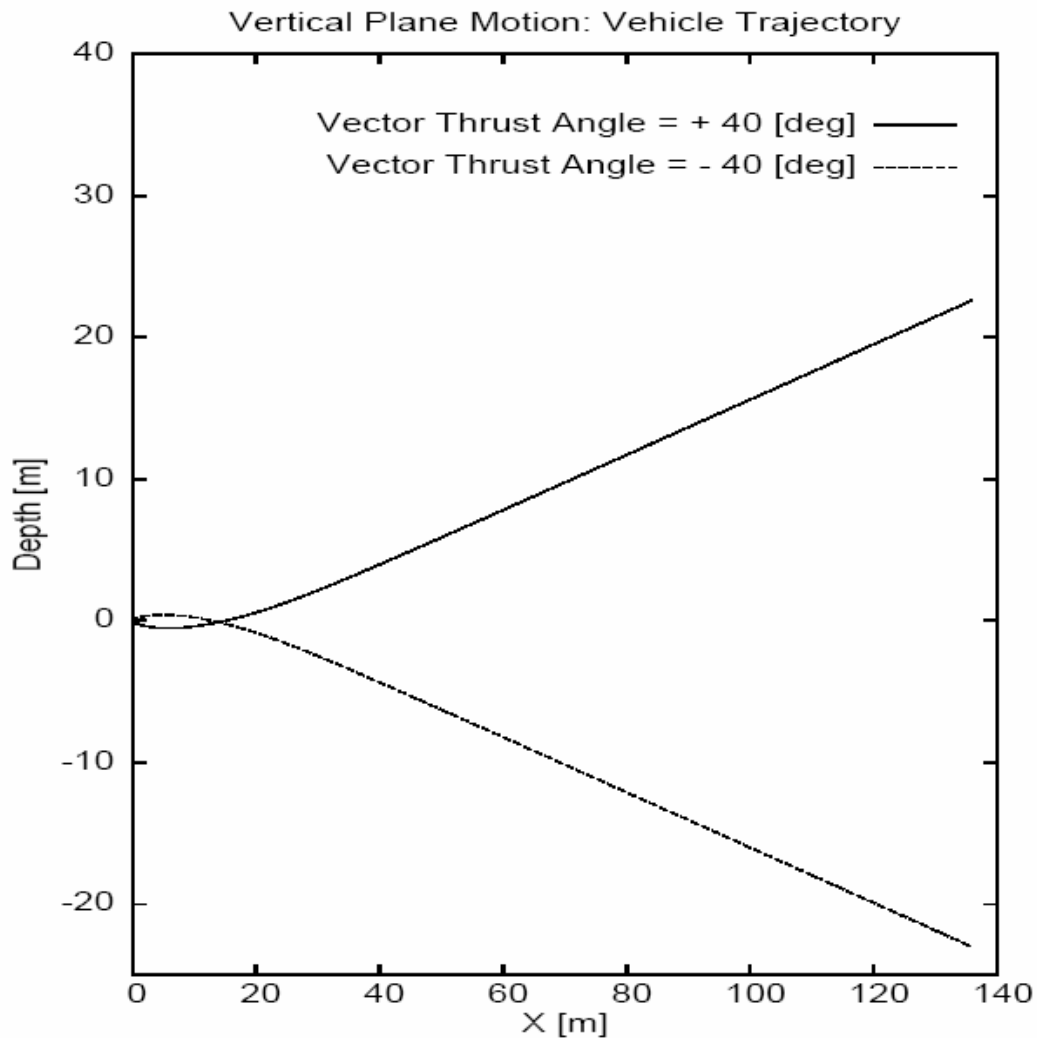


Figure 2.7.4.6 Trajectories of RPUUV with fins for thruster angle  $\beta = +40$  [deg] (solid line) and  $-40$  [deg] (dashed line)

### Findings.

Simulations of vertical plane motion of the vehicle show

- The vehicle is dynamically stable both with and without aft fins.
- Addition of fins does effect the vechile motion but the effect can be easily offset by vector thruster.
- There seems to be no special advantage of having aft fixed fins for the present vehicle, as maneuvering can be easily accomplished using vector thruster.

### 2.7.4.3 Vertical Plane Motion with Mast and Modem

For acoustic communication from land to RPUUV, a cylindrical mode transducer of diameter 2.5 [inch] and height 2.5 [inch] was mounted on top forward of the vehicle as show in in Figure 2.7.1.1. For vehicle identification in murkier waters, a slender mast of diameter 1/8 [inch] and height of 5 [ft] was mounted on the vehicle (see Figure 1.1). The presence of modem and mast introduces considerable drag force. The vector thruster chosen for the vehicle has enough power to overcome the additional drag. The effect of the pitch moment of the drag force on mast and modem is thoroughly examined in this section. All the simulations are carried out for a duration of 100 [s] with the vehicle started from rest. The thruster is ramp started from zero thrust at  $t=0$  [s] to specified thrust at  $t = 30$  [s].

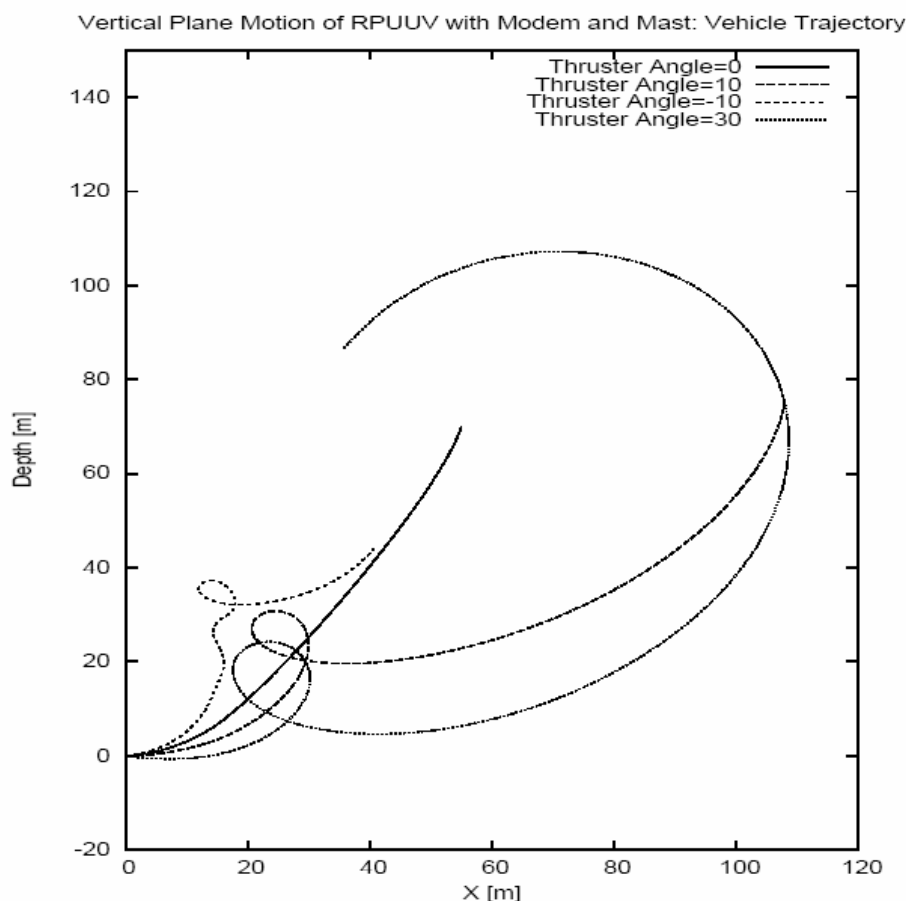
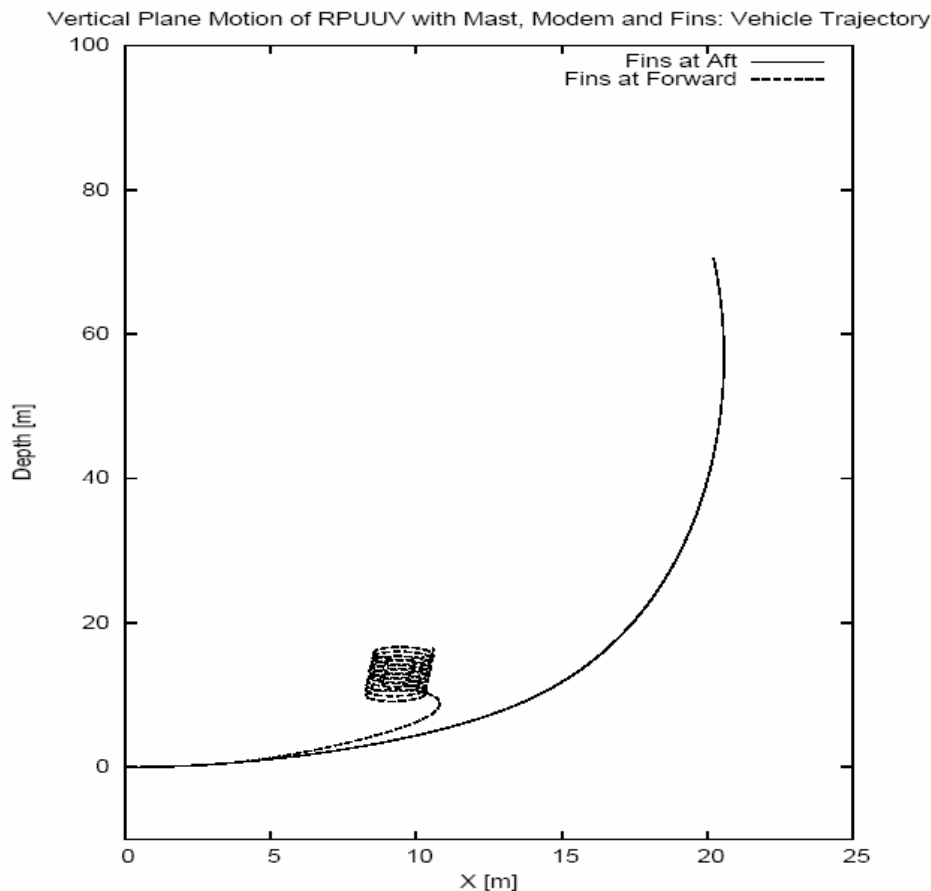


Figure 2.7.4.7 Trajectories of RPUUV with modem and mast and without fins for various thruster angles

The vehicle trajectories in the vertical plane for various thruster angles are shown in Figure 2.7.4.7. As can be observed, the drag moments from the mast and modem makes the vehicle move upwards. The vehicle trajectories can be controlled with the thruster, but keeping the thruster angle constant (as done in the simulations) cannot make the vehicle go in a straight line forward motion. An active control of thrust angle may be required for the purpose. As it will complicate the operation of the vehicle, the active control option is not considered further.



**Figure 2.7.4.8** Effect of fins and their locations on the trajectories of RPUUV with modem and mast. Thrust = 5 [N] and thruster angle = 0 [deg].

The possibility of using fixed fins to offset the adverse effect of modem and mast is next considered. Simulations with fixed horizontal fins each of chord length 0.05 [m] and span length 0.05 [m] and the slope of the lift-coefficient curve  $dC_L/d\delta$  as 9.0 [per radian] were then carried out. The fins were located at various stations along the vehicle and the thruster angle was set at 0 [deg]. As can be observed in Figure 2.7.4.8, the fins whether they are aft or forward, makes the situation only worse. In fact the vehicle trajectory becomes so irregular, when the fins are at forward, that the present model may no longer be valid. Numerous simulations were carried out, but addition of fixed fins does not appear to be the solution to the problem caused by the addition of mast and modem.

Next, simulations were carried out **without the mast** which is the major contributor to the pitch moment. The simulations carried out on the vertical plane motion of the RPUUV with the modem, but without the mast and fins, are shown in Figure 2.7.4.9. The vehicle is quite stable and controllable. At thruster angle of 0 [deg], the vehicle tends to rise because of the drag moment of the modem. But this can be easily offset using the vector thruster. As small as 1 [deg] thrust angle is enough to make the vehicle go more horizontal overcome the drag moment of the modem. With thruster angle of 2 [deg], one can even make the vehicle dive down.

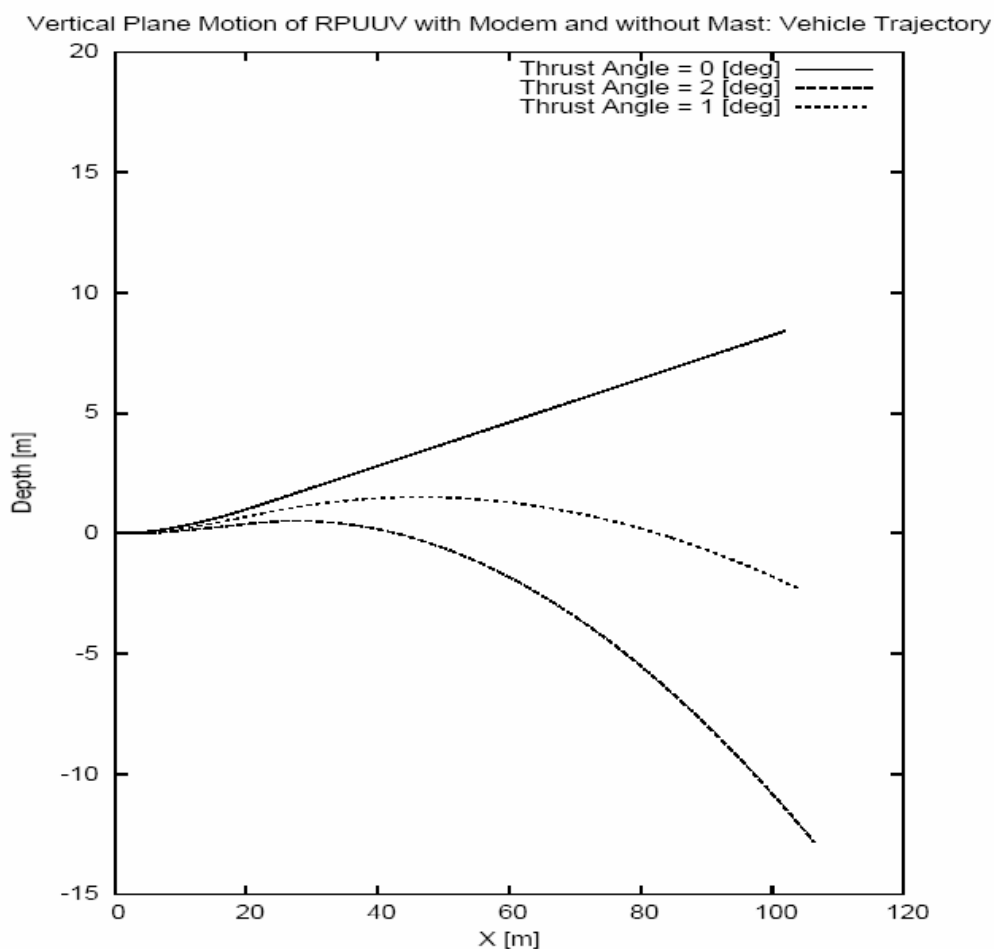


Figure 2.7.4.9 Vehicle motion without mast and fins but with modem for thrust = 5 [N] at thrust angles 0 [deg], 1 [deg] and 2 [deg].

### **Findings.**

Based on the above simulations of the vehicle motion involving mast, modem and fins, we conclude the following:

- The vehicle motion is adversely affected by the drag force moment of the mast. The pitch motion induced by the moment cannot be controlled, passively, by either the vectored thruster or the fins.
- The motion induced by the drag force and moment of the modem transducer can be easily handled by the vectored thruster.
- The passive fins do not improve the vertical-plane motion performance of the vehicle with appendages such as modem and mast.
- In the Year 1 work, we have considered the effect of tow-float cable tension on the vehicle motion. The longitudinal location of the tow-cable connection was more crucial. By connecting the tow cable to the middle of the vehicle, one can minimize the pitch oscillations caused by the cable tension.
- There is no analogous and practical solution to the instability caused by the mast. The drag force acting on the mast induces a large pitch moment and destabilizes the vehicle.
- Introducing a counter mast on the bottom is a possibility but not a practical solution. One could consider moving the center of gravity so much forward as to generate a righting moment to counter the pitch moment of the mast drag; again this is not a practical solution. The use of mast is therefore not recommended.

#### **2.7.4.4 RPUUV Motion Over Sea Bottom**

Finally, we present results for the RPUUV motion above the sea bottom. Both hydrodynamic and dynamic simulations were carried out for various altitudes of operation above the sea bottom. The values of principal hydrodynamic coefficients corresponding to unsteady vehicle motion are given in Table 4.1. The values for operation in infinite fluid, vehicle axis 0.425 [m] is above bottom and vehicle axis 0.085 [m] above bottom are given. As can be observed, the bottom effect is negligible if the vehicle is operating at  $h = 0.425$  [m] which is one-half body length above the bottom. However, when the vehicle is very close to the bottom ( $h=0.085$  [m]), the bottom effect on hydrodynamic coefficients becomes significant.

Table 2.7.4.1 Unsteady hydrodynamic coefficients for various altitudes above sea bottom.

Coefficient	Infinite fluid	$h = 0.425$ [m]	$h = 0.085$ [m]
$\mu_{11}$	0.48 [kg]	0.49 [kg]	0.64 [kg]
$\mu_{22}$	11.6 [kg]	11.7 [kg]	16.1 [kg]
$\mu_{66}$	0.681 [kg-m <sup>2</sup> ]	0.682 [kg-m <sup>2</sup> ]	0.992 [kg-m <sup>2</sup> ]

The proximity to the bottom will restrict, obviously, vehicle motions in the vertical plane. To determine bottom effect on horizontal plane motion, simulations were carried out results of which are discussed in the following paragraphs.

Trajectories of the vehicle in infinite fluid and when 0.085 [m] above the sea bottom are given in Figure 2.7.4.10. As one can observe, the bottom effect on the horizontal plane motions is rather negligible. With vector thrust applied at 10 [deg], the vehicle executes a circular path of radius of about 5 [m] in both cases.

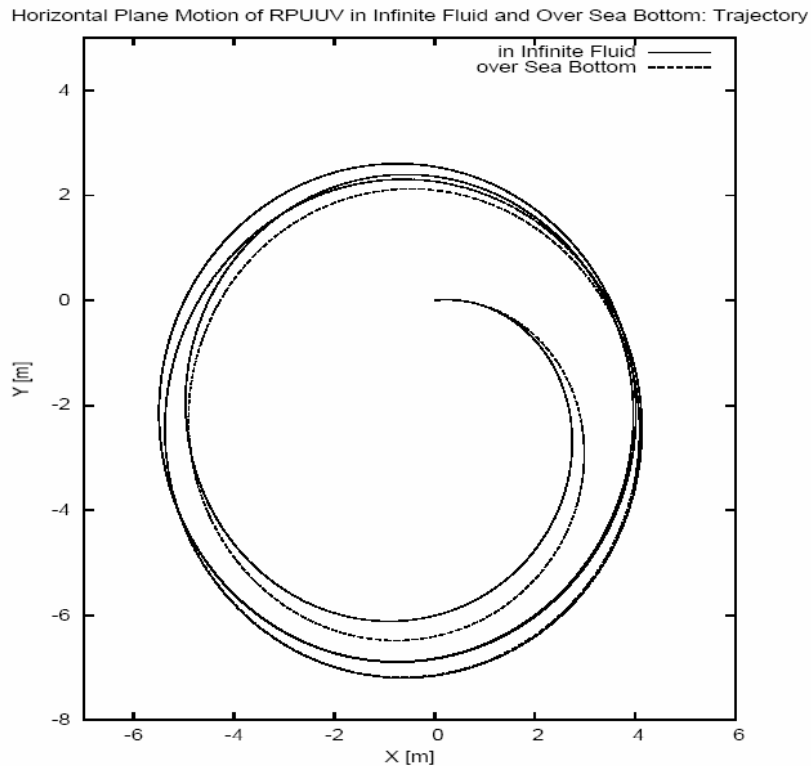


Figure 2.7.4.10 Horizontal plane motion in infinite fluid and over sea bottom ( $h=0.085$  m) both with thrust = 5 [N] and thrust angle = 10 [deg].

With fixed fins in the aft, the vehicle executes a turn as shown in Figure 2.7.4.11. Again, the sea bottom effect is quite negligible. Finally, the dynamic stability of the vehicle when operating close to the bottom is studied.

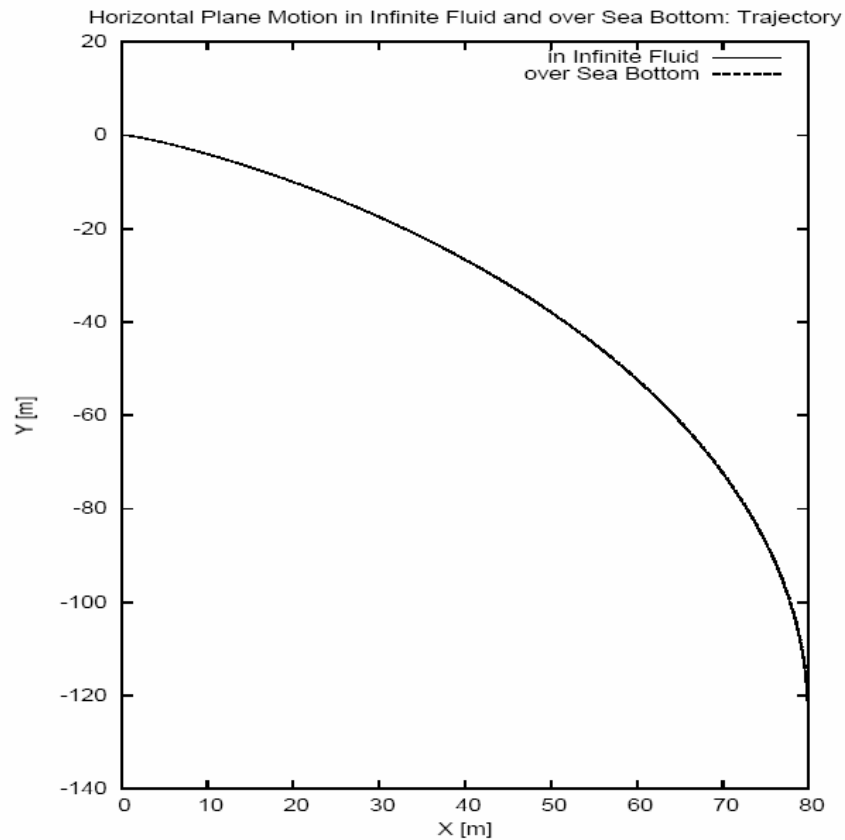


Figure 2.7.4.11 Horizontal plane motion in infinite fluid and over sea bottom ( $h=0.085$  m) both with thrust = 5 [N] and thrust angle 10 [deg] and aft fixed fins.

A perturbation in velocity is introduced at time  $t = 50$  [s] when the vehicle is operating 0.085 [m] above ground. As can be seen in Figure 2.7.4.12, the perturbation vanishes rapidly meaning the vehicle's stability is not affected by the sea bottom.

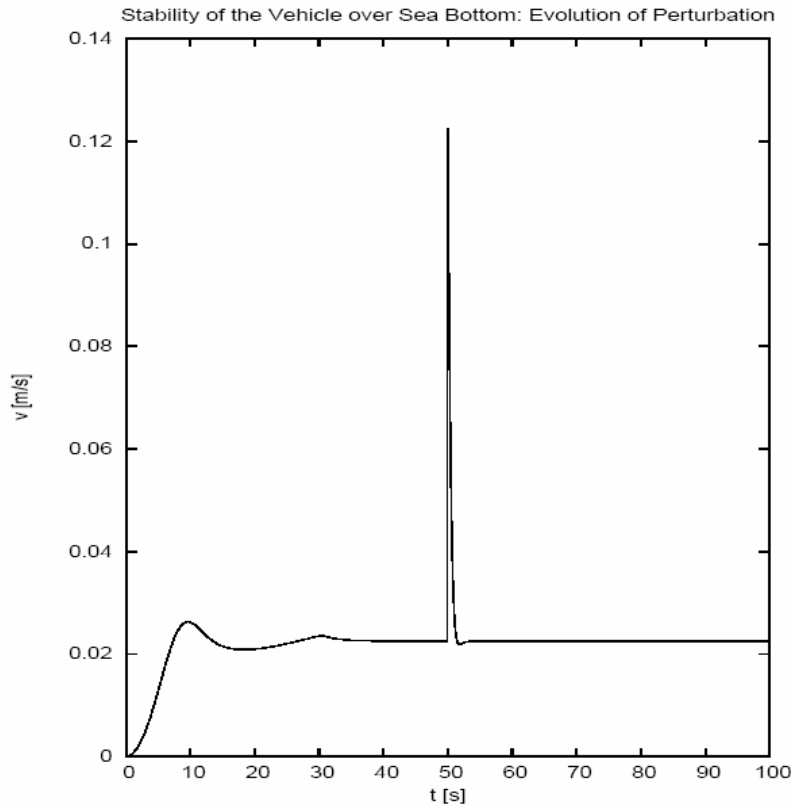


Figure 2.7.4.12 Temporal evolution of sway velocity with a perturbation introduced at  $t = 50$  [s]. The vehicle with aft fins is operating 0.085 [m] above ground with thrust = 5 [N] and thrust angle 10 [deg].

### **Findings**

Based on the simulations including sea bottom effect, we conclude

- Hydrodynamic simulations show that the sea bottom effect on hydrodynamic coefficients is significant only when the vehicle is very close to the bottom. This could be due to the three dimensionality of the vehicle which does not allow trapping of fluid underneath the vehicle.
- Dynamic simulations of horizontal motion reveal that bottom effect is very negligible.
- Any limitation to the motion in the vertical plane is only due to the actual bottom itself and not due to any change in the hydrodynamics caused by the bottom.



### 2.7.5 Conclusion

As discussed in this report, we have carried out the following investigations during Year 2 of the project

- Determination of unsteady hydrodynamic coefficients of the vehicle including sea bottom effects,
- Simulation of horizontal vehicle motion including effects of fixed fins, sea bottom and perturbations.
- Simulation of vehicle motion including appendages such as modem transducer, mast and fins.

The resulted findings and contributions to design and performance improvement are

- The effect of sea bottom of horizontal plane motion is negligible
- The instability caused by the mast cannot be suppressed passively by vectored thrust or fins. The pitch motion can be suppressed only by a counter mast on the bottom or by moving the center of gravity sufficiently further to generate required righting moment to cancel the drag moment of the mast. These solutions are not practical.
- The motion of the vehicle with modem can be easily controlled.
- Vector thruster is quite effective for maneuvering

Our Year 3 efforts will include

- Completion of the viscous flow simulation to examine geometry effects on flow separation and flow into the thruster.
- Investigation of surface wave effects

These will be presented in the Year 3 report.

## REFERENCES FOR SECTION 2.7

- [1] P. Ananthakrishnan and Sophie Decron, ``Dynamics of Small- and Mini-Autonomous Underwater Vehicles: Part I. Analysis and Simulation for Midwater Applications'', *Technical Report*, 63p, Department of Ocean Engineering, Florida Atlantic University, July 2000.
- [2] J. P. Comstock (editor), *Principles of Naval Architecture*, SNAME Publication, 1967
- [3] C. T. Crowe, D. F. Elger and J. A. Roberson, *Engineering Fluids Mechanics*, John Wiley & Sons Inc., 2005.
- [4] J. N. Newman, *Marine Hydrodynamics*, The MIT Press, 1999.
- [5] C. Puaut, ``Hydrodynamic Analysis of an Underwater Vehicle Including Free-Surface Effects'', *MS Thesis*, Advisor: P. Ananthakrishnan, Department of Ocean Engineering, Florida Atlantic University, 2001.
- [6] H. Schlichting, *Boundary Layer Theory*, McGraw-Hill, New York, 1955.
- [7] J. V. Wehausen, *Ship Dynamics*, Lecture Notes, University of California at Berkeley, Fall 1972.

## **2.8 RP UUV Navigation and Control**

**PI: Dr. Edgar An**

### **2.8.1 Summary**

To monitor coastline security during a mission, the RPUUV operators must not only analyze in real-time the video and acoustic data via the high-speed acoustic link, but also position the vehicle accurately by directly controlling the vector thruster. The latter task generally requires a great deal of effort from the operators, and thus it is highly desirable to automate the control process so that the operators can focus mostly on the data analysis and threat identification. One way to achieving this objective is to allow the operators to command using only waypoints or set points instead of controlling the vector thruster's angle and speed. The vehicle must then be capable of determining its position and attitudes accurately, and self-adjusting the thruster dynamics accordingly. Currently, there is no navigation hardware / software on the RPUUV although the vehicle is capable of receiving its USBL position fixes but at a very slow update rate. Controlling the position and attitude of the RPUUV adequately would require a much higher update rate, on the order of 10Hz.

The main objective of the proposed task is to evaluate a number of inexpensive, alternative navigation sub-systems for the remotely piloted UUV. The Year 2 achievements consist of: researching the latest navigation sensors available on the market, investigating two navigation solutions suitable for the RPUUV, and evaluating the position error performance based on 3D vehicle motion simulation and at-sea data collected using the OEX AUV from FAU.

### **2.8.2 Introduction**

Unmanned underwater vehicle (UUV) navigation has been well researched in the context of military and oceanographic tasks. Considerable success has been demonstrated using a fairly standard suite of aided navigation sensors. This consists of an inertial navigation system, aided with Global Positioning System (GPS), and Doppler velocity log (DVL) and an acoustic positioning baseline system (USBL, SBL, or LBL). However, UUV navigation around ports and harbors in the context of coastline security tasks faces additional challenges. Such an environment is typically cluttered with a vast number of varying surface vessels, lines and anchors, and this is further compounded with severe acoustic multi-paths due to the effect of very shallow water and complex bottom topography. One main objective of the Year 2 work is to determine what UUV navigation technology can be leveraged, and what is not suitable given the CCST scenarios. One main constraint on the RPUUV development is cost, and thus the selection of the navigation sensors suitable for the RPUUV must address the trade-off between the cost and performance.

### 2.8.3 Methods, Assumptions, and Procedures

#### 2.8.3.1 Navigation Sensor Considerations

Because of the systematic constraint, the navigation system must be inexpensive, and thus the investigation must be based largely on performance and cost. Typical velocity sensors are based on the well-known Doppler Velocity Log (DVL) technology, and are very expensive. As a result, this investigation primarily has considered solutions that don't require a DVL in the navigation suite. However, without any velocity measurements, a simple "speed-heading" dead reckoning solution will result in a significant amount of drift errors. Some form of position measurements is thus required to bound the drift error for position estimation. Since the RPUUV is equipped with a high-speed acoustic modem that can handle navigation and communication, this investigation thus assumes that the position measurements are available regularly (although at a slower update rate) during a mission in terms of range, elevation and azimuth angles. In terms of attitude and heading reference sensors, TCM2 (a digital compass and tilt sensor from PNI Inc.) has been the "standard" choice among most of the underwater vehicles because of its performance and cost combination, and thus other attitude and heading reference system (AHRS) sensors were not considered. The investigation was focused mainly on the inertial measurement unit (IMU) products that are available on the market, which are described in the three tables shown in Table 2.8.1-2.8.3. The important parameters of consideration in this study are the acc and gyro biases. They are highlighted in yellow color in the tables. While the products' prices are correlated to the performances, the largest acc and gyro biases are 30mg ( $\sim 0.3\text{m/s}^2$ ) and 1 deg/second. These numbers are later used as the worst-case scenario in the analysis of the navigation performance.

#### 2.8.3.2 Navigation Performance Evaluation

As position updates are available to the RPUUV on a regular basis, the navigation performance evaluation should be based on how well the vehicle navigates between fixes (i.e. interpolating its position between fixes). Since there is no velocity sensor assumed onboard, high bandwidth IMU must be incorporated in order to provide high-resolution position interpolation and velocity estimation. Underwater vehicle navigation performance is commonly evaluated using modeling and simulation because it can provide better analysis on parametric sensitivity. However, it is important to validate the assumptions made in the simulation results using experimental data. In this section, four different navigation scenarios were considered. They are:

1. Global positioning unit (GPS), Accelerometers, TCM2 using at-sea data
2. GPS, Accelerometers, Gyros, TCM2 using at-sea data
3. GPS, Accelerometers, Gyros, TCM2 using 3D simulated vehicle motion
4. USBL, Accelerometers, Gyros, TCM2 using 3D simulated vehicle motion

The first two cases involve using real at-sea data whereas the latter two cases involve using 3D simulated vehicle motion data. In the first two cases, GPS measurements were

used as position measurements. This was chosen because the vehicle was programmed to run in a straight-line mission on the surface. In the mission, USBL was not collected, and thus GPS was used as a substitute. Unlike that in Case #1, gyro data were considered in all the other cases. Case #4 will reveal the filtering performance between using USBL instead of GPS measurements.

### **2.8.3.3 Review of an Error State Kalman Filter Without Gyros**

Consider a computed velocity quantity,  $\hat{v}$ , based on integrating the accelerometer measurements. Also, assume that the AHRS sensor is perfect, thus  $C_b^n$  is noiseless. Define an error state quantity,  $e_v$ , which is part of the filter.

Company model	Cloud Cap Technology	O-NAVI	BAE Systems	Inertial Science	Inertial Science
website	Crista www.cloudcaptech.com	Falcon GX www.o-navi.com	SilMU 01 www.baesystems.com	DMARS-I-B www.inertialscience.com	ISIS-IMU www.inertialscience.com
Operating temp (C)	-40 to 80	-40 to 85	-40 to 75	-20 to 55	-40 to 70
operating voltage (V DC)	4.4 to 8	4.75 to 5.25	+/-15V, 5V	17 to 40	N/A
power (W)	< 0.72W	445mW (nom)	5	22	< 6
cost \$					
weight	37g	N/A	250g	5.5lb	< 0.8lb
size	2.05" x 1.55" x 1"	63.5mm x 31.75mm x 15.62mm	45.5mm x 80mm (dia)	4.3" dia x 4.5"	2.8" x 3" x 2.3"
output format	RS232 or CAN	RS232	RS232	RS422	RS232/RS422
<b>Accelerometer Specifications</b>					
number of axis	3	3	3	3	3
range (g)	+/- 10	+/- 2	+/- 50	+/- 6	+/- 20
Bias (mg)	< 25mg (fixed temp)	N/A	0.5	0.2	2
Scale factor accuracy	< 1%	N/A	2000ppm	200ppm	< 0.2%
non linearity (%F.R)	N/A	0.2%		N/A	< 0.5%
resolution (mg)	0.3mg	N/A		N/A	1
Update rate (Hz)	> 200	50	100	50	> 30
random walk (m/s/sqrt(hr))	N/A	N/A	1	N/A	< 0.1
output noise	< 12mg	200ug/sqrt(Hz)		10ug/sqrt(Hz)	N/A
alignment error	2.6% (uncorrected)	1 deg	1mrad	75urad	< 2 mrad
<b>Gyro Specifications</b>					
number of axis	3	3	3	3	3
range (deg/sec)	+/- 300	+/- 150	+/- 600 (x,y), +/- 1000 (z)	+/- 200	+/- 90
bias (deg/sec)	< 0.2 (fixed temp)	N/A	5 deg/hr	< 0.3 deg/hr	0.01
scale factor accuracy	< 1%	N/A	1500ppm	200ppm	< 0.2%
nonlinearity (%F.S.)	N/A	0.1%		N/A	< 0.2%
resolution (deg/sec)	0.009	N/A		N/A	1 arc secs / sec
bandwidth (Hz)	> 200	40	100	50	> 30
Random walk (deg/sqrt(hr))	N/A	0.05	1	0.005	< 0.5

Table 2.8.1: A comparison of existing IMU product specifications

Company	MemSense	MemSense	MemSense	KVH	BEI	BEI
model	uIMU	nIMU	AccelRate3D (AR02)	TG-6000	DQI	MMQ 50
website	www.memsense.com	www.memsense.com	www.memsense.com	www.kvh.com	www.syston.com	www.syston.com
Operating temp ( C )	0 to 70	0 to 70	-40 to 85	-20 to 55	-54 to 71	-54 to 70
operating voltage (V DC)	9 to 13	5.4 to 9	4.75 to 5.25	14 to 30	28	+/- 12
power (W)	N/A	N/A	N/A	< 12	13	< 5
cost \$						
weight	150g	15g	5g	< 4lb	2.2lb	< 0.5lb
size	N/A	N/A	0.7" x 0.7" x 0.403"	5.63" x 5.13" x 4.55"	3.189" x 3.53" x 3.84"H	1.88" x 1.88" x 2.55"
output format		RS422/I2C	analog	RS422	AMRAAM serial	RS232
Accelerometer Specifications						
number of axis	3	3	3	3	3	3
range (g)	+/- 2	+/- 2	+/- 2	+/- 70	+/- 70	+/- 10
Bias (mg)	30	30	N/A	10	1.5	< 3
Scale factor accuracy	N/A	N/A	N/A	0.5%	350ppm	< 5000ppm
non linearity (%F.R)	< 1	< 1	< 1	2000ppm	N/A	N/A
resolution (mg)	N/A	N/A	1000mV/g	0.3	N/A	N/A
Update rate (Hz)	50	50	50	150	N/A	50
random walk (m/s/sqrt(hr))	N/A	N/A	N/A	N/A	N/A	N/A
output noise	7.5mg	7.5mg	< 65 ug / sqrt(Hz)	8mg	200ug / sqrt(Hz)	0.5mg / sqrt(Hz)
alignment error	1%	1%	2%	0.4 deg	0.5mrad (1 sigma)	< 5mrad
Gyro Specifications						
number of axis	3	3	3	3	3	3
range (deg/sec)	+/- 300	+/- 300	+/- 150	+/- 750	+/- 1000	200
bias (deg/sec)	1 deg/sec	1 deg/sec		10 deg/hr	10 deg/hr	50 - 200 deg/hr
scale factor accuracy	N/A	N/A		2500ppm	350ppm	< 5000ppm
nonlinearity (%F.S.)	0.1	0.1	0.1%	1000ppm	N/A	N/A
resolution (deg/sec)	N/A	N/A	12.5 mV/deg/s	0.03	N/A	N/A
bandwidth (Hz)	50	50	50	150	N/A	50
Random walk (deg/sqrt(hr))	N/A	N/A	0.05	0.1	0.035	0.3

Table 2.8.2: A comparison of existing IMU product specifications

Company	IMAR	Crossbow	Crossbow	Crossbow	Honeywell	Lifton
model	iNav-FMS T	IMU300CC-100	IMU400CD-100	IMU700CB-200	HG1700 AG58	LN 200
website	www.imar-navigation.de	www.xbow.com	www.xbow.com	www.xbow.com	www.honeywell.com	www.nsd.es.northropgrumman.com
Operating temp (C)	N/A	-40 to 85	-40 to 71	-40 to 60	-54 to 85	-54 to 71
operating voltage (V DC)	10 to 34	9 to 30	9 to 30	10 to 30	N/A	N/A
power (W)	10	< 3	< 3	< 8	< 8	12 (steady state)
cost \$		\$1995				
weight	1.7kg	< 1.3lb	< 1.3lb	< 3.5lb	< 1.6lb	< 1.65lb
size	265 x 145 x 132 (mm)	3" x 3.2" x 3.75"	3" x 3.2" x 3.75"	5" x 6" x 4"	3" x 5" (dia)	3.5" (dia) x 3.35"
output format	RS232	RS232	RS232	RS232	RS422	RS485
Accelerometer Specifications						
number of axis	3	3	3	3	3	3
range (g)	+/- 5	+/- 2	+/- 4	+/- 4	+/- 50	> 70
Bias (mg)	< 1	< +/- 30	< +/- 12	< +/- 12	1	300 ug to 3mg (1 sigma)
Scale factor accuracy	N/A	< 1%	< 1%	< 1%	300ppm	300 to 500ppm (1 sigma)
non linearity (%F.R)	< 500 ppm	< 1	< 1	< 1	500ppm	N/A
resolution (mg)	< 100 ug	< 1	< 0.6	< 0.6	N/A	N/A
Update rate (Hz)	N/A	> 75	> 75	> 75	100	N/A
random walk (m/s/sqrt(hr))	N/A	< 0.15	< 1	< 1		N/A
output noise	N/A	N/A	N/A	N/A	85ug / sqrt(Hz)	N/A
alignment error	N/A	N/A	N/A	N/A	21 arc sec (1 sigma)	N/A
Gyro Specifications						
number of axis	3	3	3	3	3	3
range (deg/sec)	+/- 500	+/- 100	+/- 100	+/- 200	+/- 1000	+/- 11459
scale factor accuracy	0.75 deg/hr	< +/- 2	< +/- 1	< +/- 20(deg/hr)	1 deg/hr	1 to 10 deg / hr (1 sigma)
nonlinearity (%F.S.)	N/A	< 1%	< 1%	< 2% over temp	150ppm (1 sigma)	100 to 500ppm (1 sigma)
resolution (deg/sec)	< 300 ppm	< 0.3	< 0.3	< 1	N/A	N/A
bandwidth (Hz)	N/A	< 0.025	< 0.025	< 0.025	N/A	N/A
Random walk (deg/sqrt(hr))	< 0.1	< 2.25	< 2.25	< 0.4	100	N/A
						0.07 to 0.15

Table 2.8.3: A comparison of existing IMU product specifications



$$v^n = v_0^n + \int_0^t a^n dt = v_0^n + \int_0^t C_b^n a^b dt$$

$$\hat{v}^n = \hat{v}_0^n + \int_0^t C_b^n \tilde{a}^b dt = \hat{v}_0^n + \int_0^t C_b^n (a^b + \nabla + n_a) dt$$

$$e_v = v^n - \hat{v}^n = e_{v_0} - \int_0^t C_b^n (\nabla + n_a) dt$$

$$\rightarrow \dot{e}_v = -C_b^n (\nabla + n_a)$$

$$\rightarrow \dot{e}_x = e_v$$

Define an error state position quantity,  $e_x$ , as  $x - \hat{x}$ , where

$$\hat{x} = \hat{x}_0 + \int_0^t \hat{v} dt$$

The error measurement quantity is defined as

$$\tilde{x} = x + n_{GPS}$$

$$e_{x_m} = \tilde{x} - \hat{x} = x + n_{GPS} - \hat{x} = e_x + n_{GPS}$$

Formulate a Kalman filter as

$$x = Ax + Gw$$

$$y = Cx + v$$

where  $x = [x \quad v \quad a \quad e_x \quad e_v \quad \nabla]^T$ ,  $y = [\tilde{x} \quad e_{x_m} \quad \tilde{a}]^T$ ,  $w = [\Phi \quad \Phi \quad n_a \quad \Phi \quad n_a \quad \Phi]^T$ ,

and  $v = [n_{GPS} \quad n_{GPS} \quad n_a]^T$

$$A = \begin{bmatrix} \Phi & I & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & C_b^n & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & I & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & -C_b^n \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \end{bmatrix}, G = \begin{bmatrix} \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & I & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & -C_b^n & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \end{bmatrix}$$

$$C = \begin{bmatrix} I & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & I & \Phi & \Phi \\ \Phi & \Phi & I & \Phi & \Phi & I \end{bmatrix}$$

### 2.8.3.4 Review of an Error State Kalman Filter With Gyros

In this case,  $\tilde{C}_b^n$  is based on AHRS measurements.

$$\begin{aligned}
 v^n &= v_0^n + \int_0^t a^n dt = v_0^n + \int_0^t C_b^n a^b dt \\
 \hat{v}^n &= \hat{v}_0^n + \int_0^t \tilde{C}_b^n \tilde{a}^b dt = \hat{v}_0^n + \int_0^t \tilde{C}_b^n (a^b + \nabla^b + n_a^b) dt \\
 e_v &= v^n - \hat{v}^n = e_{v_0} - \int_0^t \tilde{C}_b^n (\nabla^b + n_a^b) dt \\
 \rightarrow \dot{e}_v &= -\tilde{C}_b^n (\nabla^b + n_a^b) \\
 \rightarrow \dot{e}_x &= e_v
 \end{aligned}$$

Note that the AHRS bias is unobservable because we don't have the velocity measurements. Thus, we assume that  $\int_0^t C_b^n a^b dt \approx \int_0^t \tilde{C}_b^n a^b dt$

To estimate the gyro bias  $\varepsilon$ , we note that

$$\begin{aligned}
 \tilde{\Omega}_{ib}^b &= \Omega_{ib}^b + \varepsilon + \eta_g \\
 \dot{E} &= J(E) \Omega_{ib}^b \\
 \dot{\tilde{E}} &= J(\tilde{E}) \tilde{\Omega}_{ib}^b \\
 \dot{e}_E &= \dot{E} - \dot{\tilde{E}} = -J(\tilde{E})(\varepsilon + \eta_g)
 \end{aligned}$$

Similarly, we assume  $J(\tilde{E}) \approx J(E)$

To incorporate these changes into the filter,

$$\begin{aligned}
 x &= Ax + Gw \\
 y &= Cx + v
 \end{aligned}$$

$$\begin{aligned}
 \text{where } x &= [x \quad v \quad a \quad e_E \quad \varepsilon \quad e_x \quad e_v \quad \nabla]^T, \quad y = [\tilde{x} \quad \tilde{a} \quad e_{E_m} \quad e_{x_m}]^T, \\
 w &= [\Phi \quad \Phi \quad n_a \quad n_g \quad \Phi \quad \Phi \quad n_a \quad \Phi]^T, \quad \text{and } v = [n_{GPS} \quad n_a \quad n_c \quad n_{GPS}]^T
 \end{aligned}$$

$$A = \begin{bmatrix} \Phi & I & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \tilde{C}_b^n & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & -J(\tilde{E}) & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & I & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & -\tilde{C}_b^n \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \end{bmatrix}, G = \begin{bmatrix} \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & I & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & -J(\tilde{E}) & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & -\tilde{C}_b^n & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \end{bmatrix}$$

$$C = \begin{bmatrix} I & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & I & \Phi & \Phi & \Phi & \Phi & I \\ \Phi & \Phi & \Phi & I & \Phi & \Phi & \Phi & \Phi \\ \Phi & \Phi & \Phi & \Phi & \Phi & I & \Phi & \Phi \end{bmatrix}$$

## 2.8.4 Results and Discussion

### Case #1: No DVL Measurements, Only Acc Bias Estimation

**Matlab Scripts:** errorStateNoDVL.m, errorKalmanNoDVL.m

Based on the description in 2.8.3.3, an error state Kalman filter was used to estimate both the acc bias and velocity of the vehicle during an at-sea surface mission. In the mission, the OEX headed west for 500 seconds, and then headed east for another 500 seconds. While it was on the surface, GPS, DVL, IMU and TCM2 data were recorded. Note that the DVL data were not used in the filter, but only used to evaluate the accuracy of the velocity estimation.

Without any filtering and bias estimation, the pure dead-reckoning solution based on IMU and TCM2 is shown in Figure 2.8.1. One can see that the acc biases are significant that without any compensation the solution is unusable.

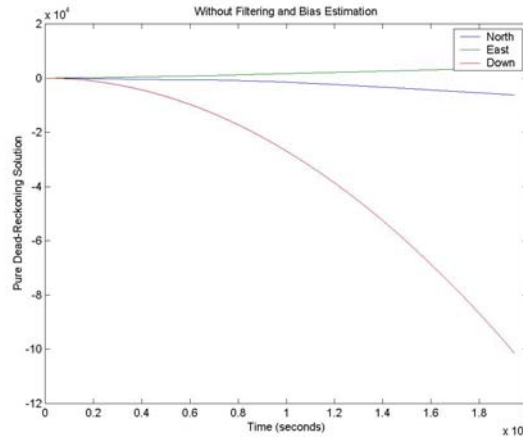


Figure 2.8.1: Dead reckoning drift due to accelerometer bias (Case #1).

With error state filtering, Figure 2.8.2 shows the convergence of the acc biases. One can see that there is a large z-component bias ( $-0.182\text{m/s}^2$ ), and its convergence was very fast. Figure 2.8.3 below shows the accuracy of the velocity estimation. One can see that the general profile of the velocity has been captured although there is a noticeable amount of noise in the estimates. Figure 2.8.4 below shows the difference between GPS and the position estimates. The GPS position update was about 1Hz. One can see from the error plots that the horizontal errors are bound mostly  $\pm 2$  meters.

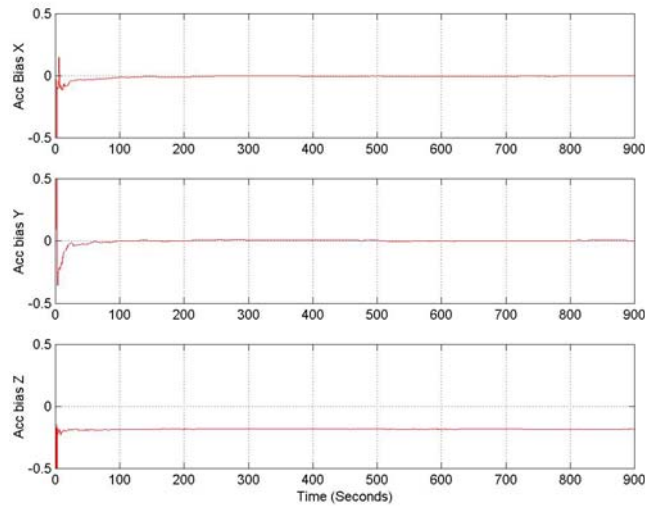


Figure 2.8.2: Acc bias estimation performance (Case #1).

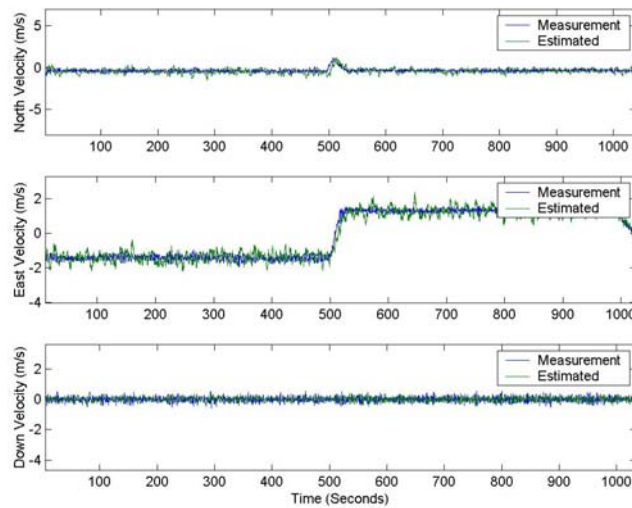


Figure 2.8.3: Velocity estimation performance (Case #1).

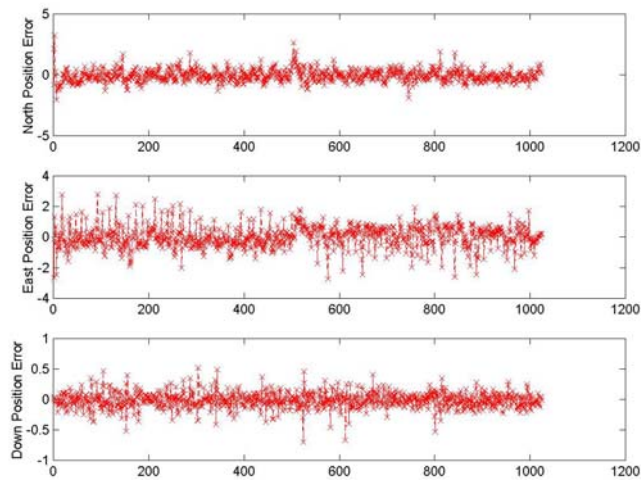


Figure 2.8.4: Position estimation performance (Case #1).

To approximate a typical acoustic position update around 0.1Hz, the GPS fixes were artificially sub-sampled to that rate, and the filter was re-run with the sub-sampled GPS data set. Figure 2.8.5 show the degradation due to a slower position update.

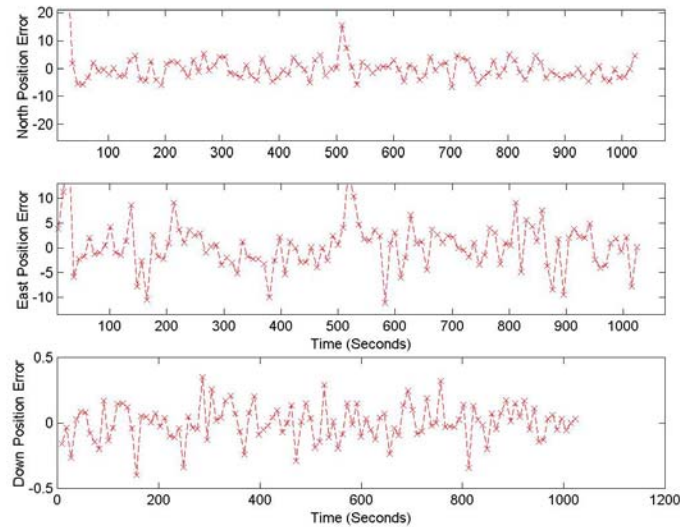


Figure 2.8.5: Position estimation performance with slower GPS fixes (Case #1).

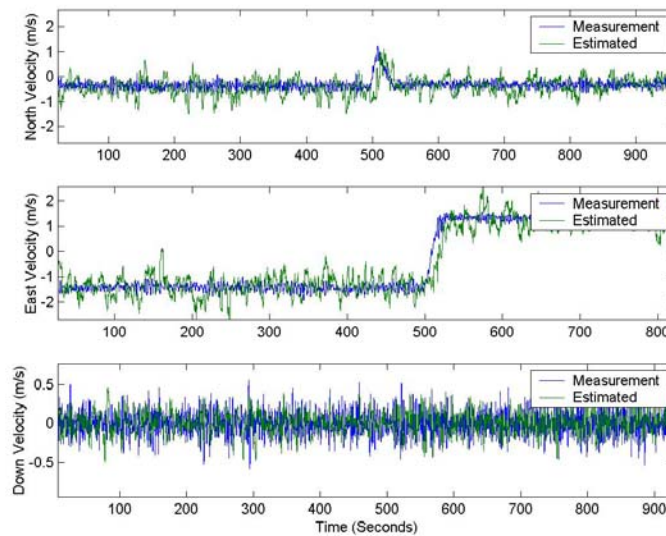


Figure 2.8.6: Velocity estimation performance with slower GPS fixes (Case #1).

Regarding the position error figure, the position estimation errors are computed as the difference between the position estimates and the original GPS data set (without sub-sampled, and are shown by 'x'). The position estimates are evaluated at the time steps right before every new GPS fixes are available. One can see from the horizontal error plots that they are bound to within  $\pm 10$  meters. The velocity estimates are much noisier with the sub-sampled GPS data set, compared to the previous case with 1Hz position update (Figure 2.8.6).

### Case #2: No DVL Measurements, Acc and Gyro Bias Estimation

**Matlab Scripts:** errorStateNoDVL2.m, errorKalmanNoDVL2.m

Based on the description in 2.8.3.4, a modified error state Kalman filter was used not only to estimate both the acc bias and velocity of the vehicle during a surface mission, but also the gyro bias. The same data set was used for the estimation as in the previous section. One can see the acc bias and velocity estimation remain the same. Figure 2.8.7 shows the magnitudes are on order of 0.05 degree per second although the convergence is not asymptotic. Figure 2.8.8 shows the measured (in red) and estimated (in blue) roll, pitch and heading angles. One can see that they are somewhat similar, and the estimated curves are smoother. Note that no attempt was made to choose the optimal estimation parameters in this case.

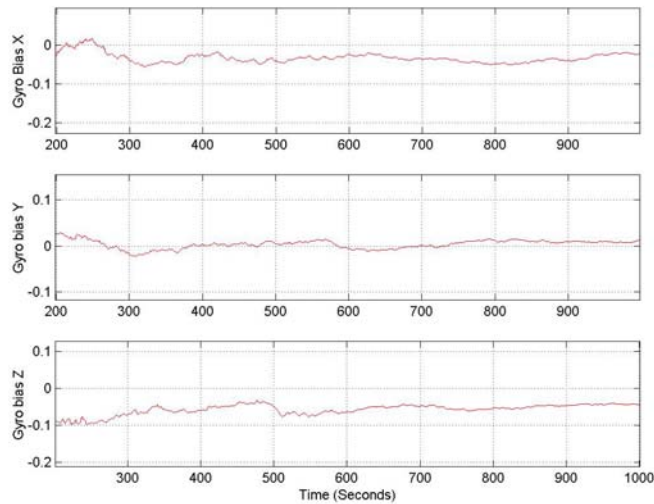


Figure 2.8.7: Gyro bias estimation performance (Case #2).

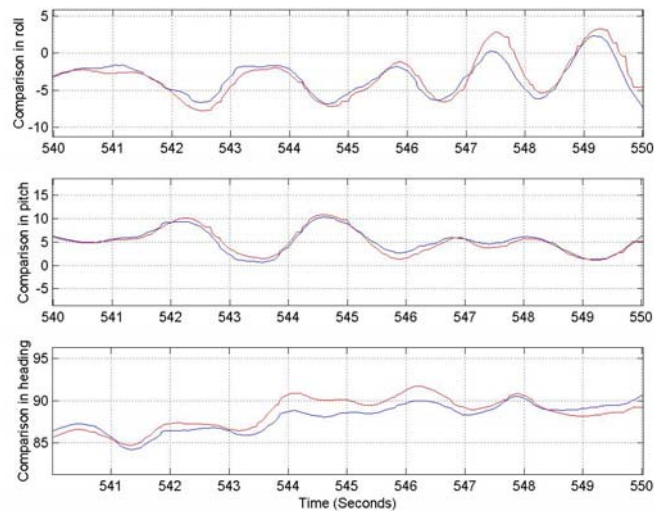


Figure 2.8.8: Comparison of TCM2 measurements and their estimates (Case #2).

### Case #3 (3D Simulated Measurements)

**Matlab Scripts:** simMainNoDVL.m, KalmanNoDVL.m (in C:\ean\Research\AUV Navigation\USBL Study)

Analysis of navigation performance using real at-sea measurement data is generally very difficult in that all available information is in form of measurements that contain noise. It is thus very difficult to verify the effect of individual parameters on the estimation performance. In this case, measurements with noise were simulated for 3D vehicle motion, and thus an exact ground truth model is available. The objective of this study is to evaluate the difference in the estimation results between a simulated run and a real mission, and perhaps from those results, a better understanding about the effect of measurement noises on the estimation performance can be gained.

The simulation considered the vehicle traveling at 1m/s at a constant heading of  $0^\circ$  for 1000 seconds. The measurement noise variances are defined as

$$\sigma_{acc} = 0.1$$

$$\sigma_{gyro} = 0.3$$

$$\sigma_{TCM2} = 0.3$$

$$\sigma_{acc} = 0.1$$

$$\sigma_{GPS} = 3$$

$$\sigma_{depth} = 0.2$$

### Case #3A

The update rates for acc, gyro, TCM2, GPS are 100Hz, 100Hz, 10Hz, and 1Hz respectively. In this case, gyro and acc biases are set to 0.01 deg/sec and 0.01 m/s<sup>2</sup> respectively. The simulated TCM2 measurements are assumed for now to have zero-biases. The same EKF described in Case #2 was used in this case. Figure 2.8.9-10 show the acc and gyro biases.

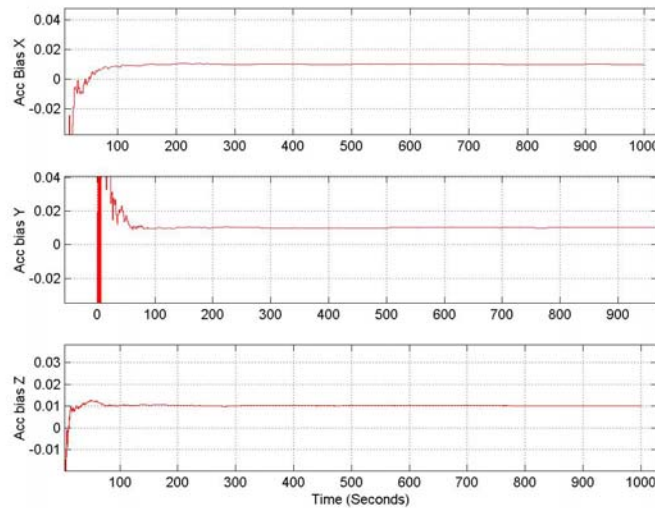


Figure 2.8.9: Simulated acc bias estimation performance (Case #3A).



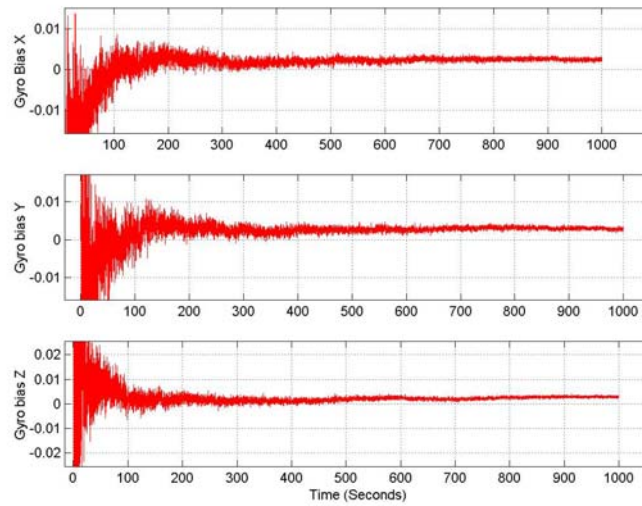


Figure 2.8.10: Simulated gyro bias estimation performance (Case #3A).

Figure 2.8.11-12 shows the position estimation error and velocity estimation performances.

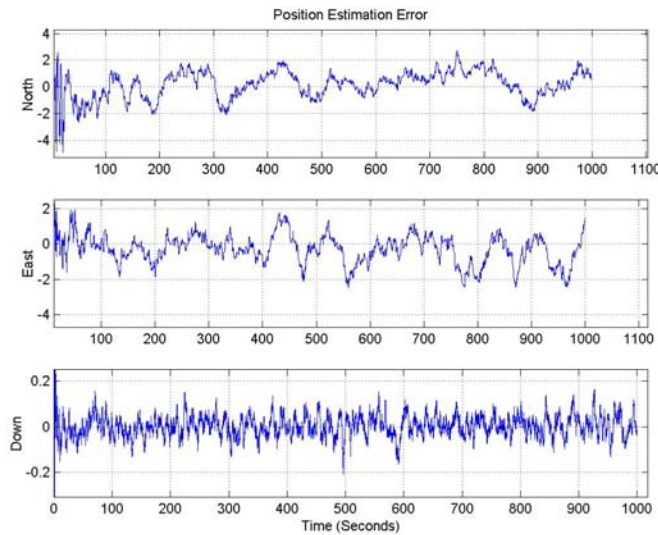


Figure 2.8.11: Simulated position estimation performance (Case #3A).

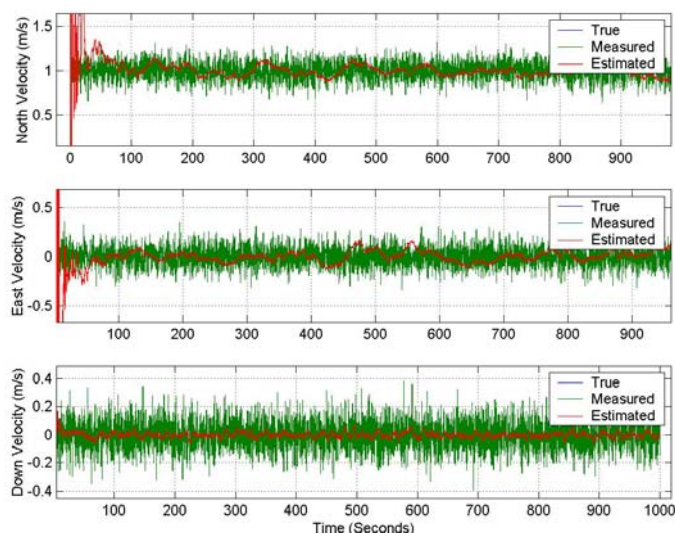


Figure 2.8.12: Simulated velocity estimation performance (Case #3A).

One can see that the position errors are  $< 2$  meters ( $x$  and  $y$ ), and  $< 0.2$  meters ( $z$ ). The velocity estimates are represented in the red curves, as compared to the simulated measurements in the green curves.

### Case #3B

Consider the GPS update rate is reduced to  $0.1Hz$ , and re-run the filter. Figure 2.8.13-14 show the velocity estimation and position estimation error performances. One can see that the position discrepancy is less than  $\pm 5$  meters ( $x$  and  $y$ ), and  $\pm 0.2$  meter ( $z$ ). The velocity estimation performance does not appear to be sensitive to the change in GPS update rate.

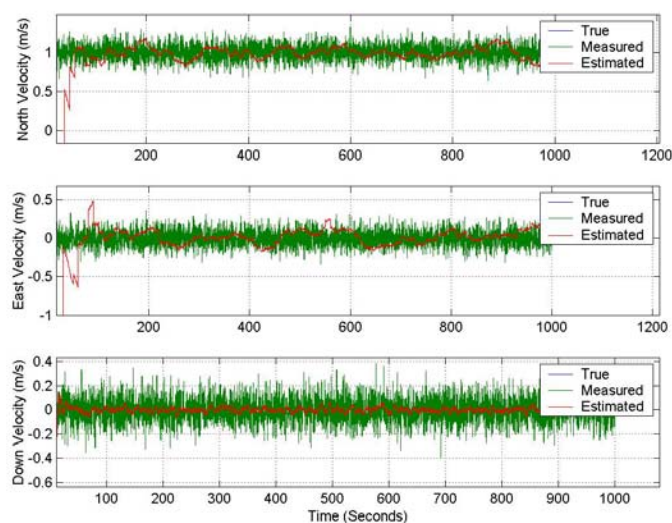


Figure 2.8.13: Simulated velocity estimation performance (Case #3B).

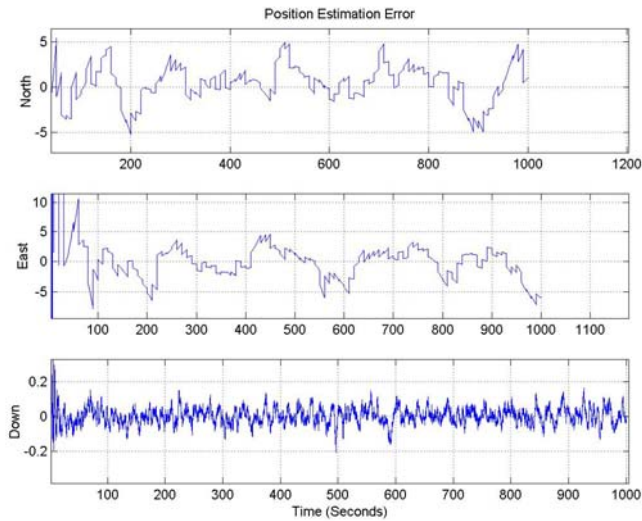


Figure 2.8.14: Simulated position estimation performance (Case #3B).

### Case #3C

Re-run the filter again except now the TCM2 biases are set to  $2^\circ$  (roll),  $2^\circ$  (pitch), and  $5^\circ$  (heading). Figure 2.8.15-16 show the velocity estimation and position estimation error performances. One can see that the position discrepancy is again less than  $\pm 5$  meters ( $x$  and  $y$ ), and  $\pm 0.2$  meter ( $z$ ). The filter appears to be somewhat robust with respect to TCM2 biases. Similarly, the velocity estimation performance does not appear to be sensitive to the change in GPS update rate.

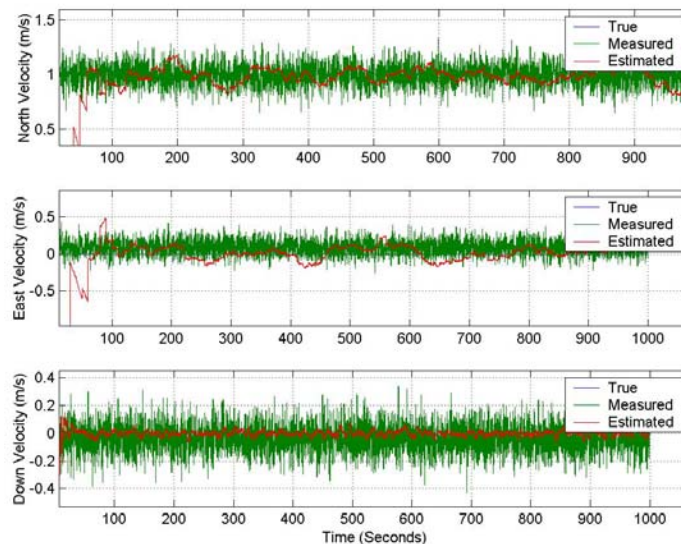


Figure 2.8.15: Simulated velocity estimation performance (Case #3C).

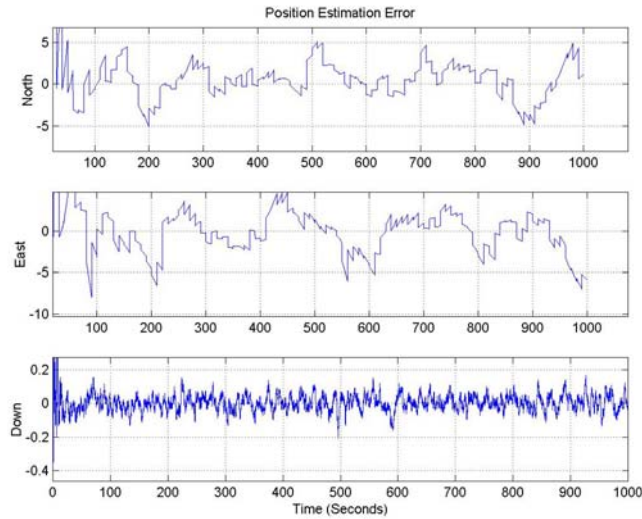


Figure 2.8.16: Simulated position estimation performance (Case #3C).

### Case #3D

Re-run the filter again except now the acc and gyro biases are set to  $0.3\text{m/s}^2$  and  $1\text{ deg/sec}$  respectively. Figure 2.8.17 and 2.8.18 show the estimation performances.

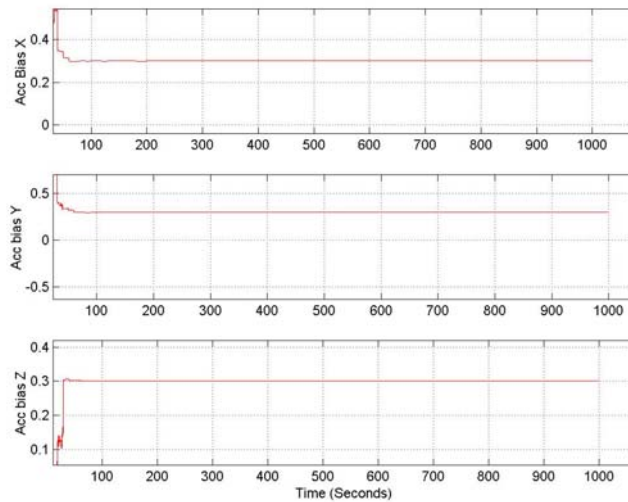


Figure 2.8.17: Simulated acc bias estimation performance (Case #3D).

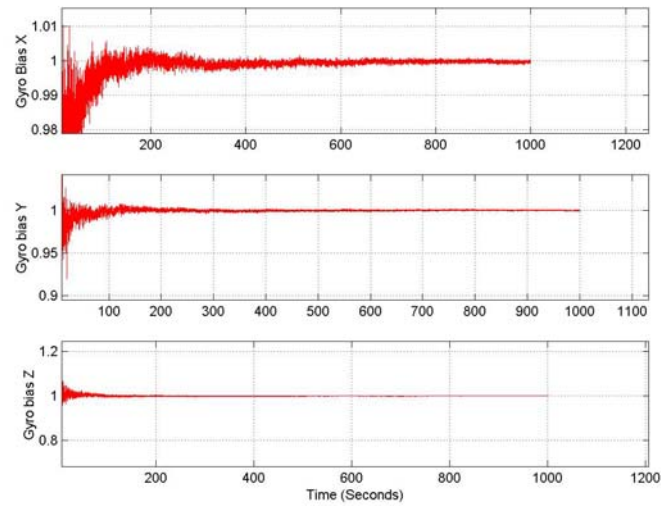


Figure 2.8.18: Simulated gyro bias estimation performance (Case #3D).

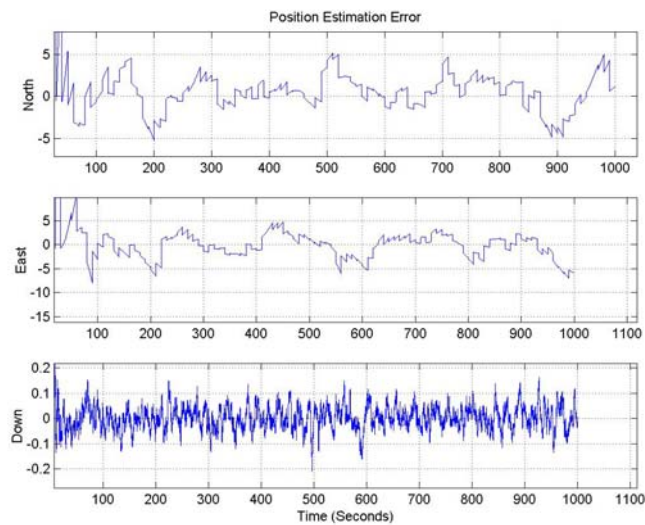


Figure 2.8.19: Simulated position estimation performance (Case #3D).



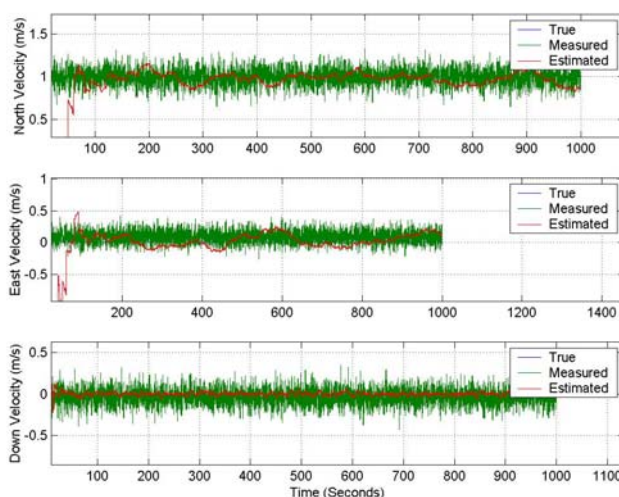


Figure 2.8.20: Simulated velocity estimation performance (Case #3D).

The results are similar to those shown in Case #3C, suggesting that the filter is capable of handling most of the COTS IMU systems. As this is a worst-case scenario for estimation, the biases are generally well less than the set values. It should be mentioned again that since it is assumed that there is no DVL sensor, it is impossible to estimate the TCM2 biases, and thus they are not shown here in the results.

#### Case #4 (USBL Range Measurements)

**Matlab Scripts:** `simMainNoDVLRange.m`, `KalmanNoDVL.m` (in C:\ean\Research\AUV Navigation\USBL Study)

In this case, the position measurements are based on a USBL sensor, instead of GPS. The simulated USBL measurements consist of range, and elevation and azimuth angles (the standard deviations are all unity). The noises associated with the angles are assumed to be of Gaussian distribution, whereas that of range is assumed to be of Raleigh distribution. These measurements are then converted to the Cartesian type via geometrical transformation. Note that the same Kalman filter is used in this case as the conversion is performed in `simMainNoDVLRange.m`. Figure 2.8.21 and 2.8.22 show the filtering performance with the USBL measurements.

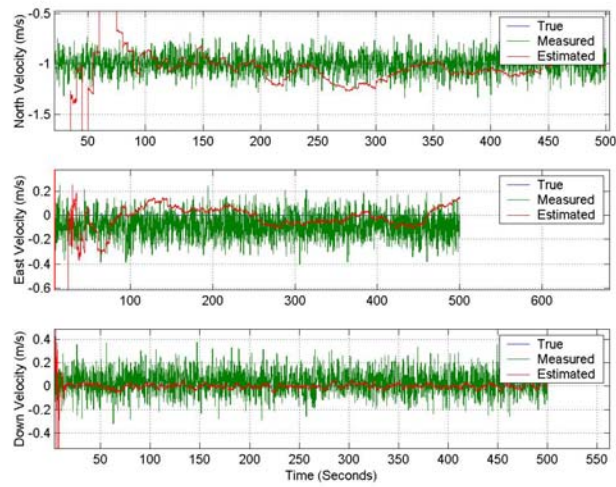


Figure 2.8.21: Simulated velocity estimation performance (Case #4).

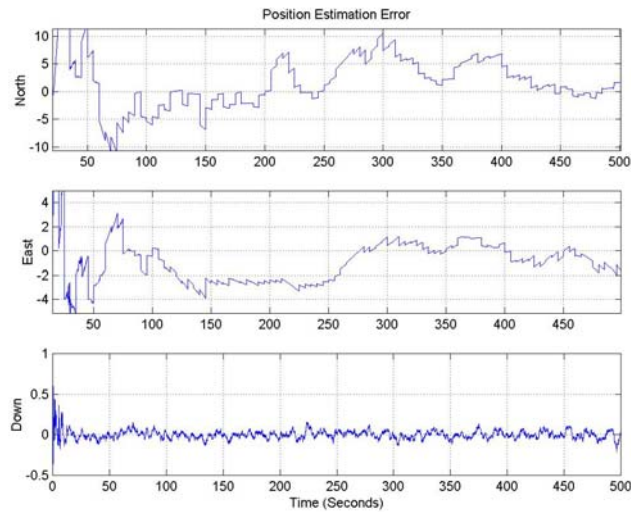


Figure 2.8.22: Simulated position estimation performance (Case #4).

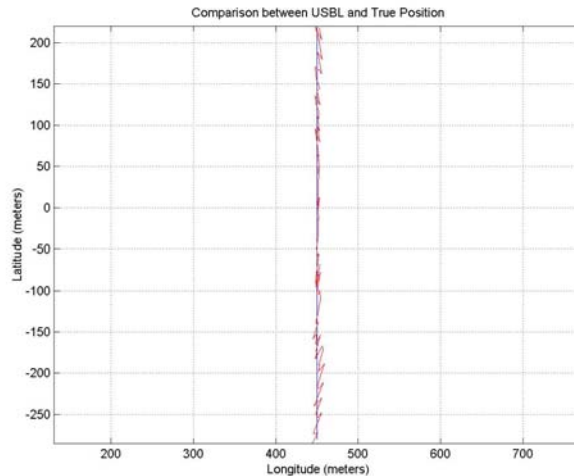


Figure 2.8.23: Simulated USBL measurement noise characteristics (Case #4).

The north and east velocity estimates are noisier as compared to those using GPS measurements. In addition, the position estimation errors are somewhat comparable to those in Case #3. It is important to point out that the range measurements are biased because of its Raleigh distribution, and the Cartesian transformation is sensitive to the range and angle measurements. This can be seen in Figure 2.8.23 in which the blue curve represents the true motion, and the red curve is the USBL-based position measurements that are fed to the filter. One can clearly see that there is large fluctuation during the beginning and the end of the run, whereas the middle portion is much less noisy. In other words, the measurements from the USBL sensor are highly time varying and range-dependent. Further investigation will be needed to determine whether the filter should be modified to account for this effect. In this particular run, the standard deviation of the USBL-based position measurements was set to 30 such that the filter should weigh much less on an instantaneous position measurement.

## 2.8.5 Conclusion

This CCST study on RPUUV navigation has been primarily focused on the selection of a cost-effective sensor suite that is appropriate to the CCST operational constraints. To provide the remote piloting capability, a high-speed acoustic modem that can handle navigation and communication will be installed on the RPUUV, and thus position measurements are available on a regular basis although the update rate will be limited to more than 1 second. One important constraint on the RPUUV is that the platform cost must be low, and thus a DVL velocity sensor was intentionally not considered because of its extraordinary cost. To provide adequate position estimation capability onboard, the position measurements must be properly interpolated, and this requires the use of IMU and AHRS sensors onboard. Four different cases were considered in this study, and the results suggest that a low-cost IMU together with a TCM2 can provide reasonable position interpolation performance between fixes, and the position errors appear to fall within  $\pm 5$  meters at a position update interval of 10 seconds, given that the vehicle travels



on the order of 1m/s. Note that in a typical port scenario, the acoustic position update interval is likely to be less than 10 seconds, resulting in further reduced position estimation errors.

### 2.8.6 Recommendations

Processing the USBL position measurements can be tricky in two aspects:

1. The noise does not follow the typical Gaussian distribution.
2. The noises in the transformed USBL position measurements are high non-stationary.

It should be noted that this study did not attempt to modify the Kalman filter in order to account for these anomalies. Despite that the position estimation performance using USBL fixes has been shown to be similar to that using GPS fixes, it is surmised that if the noise characteristics can be incorporated into the Kalman filter, the performance is expected to be better. The overall recommendation is that the navigation sensor suite should consist minimally of:

- A low-cost IMU (the noise and bias characteristics are no worse than the worst case scenario considered in this study)
- A TCM2 AHRS compass

### 2.8.7 References

- [ 1 ] P.E. An, T. Healey, On Line Compensation of Heading Sensor Bias for Low Cost AUVs, AUV 98 Navigation Workshop, pp.35-42, Draper Lab, Massachusetts, August, 1998.
- [ 2 ] J. Bellingham, S. Willcox, Optimizing AUV Oceanographic Surveys, AUV 96 Conference, pp.391-398, Monterey, CA, 1996.
- [ 3 ] Introduction to Random Signals and Applied Kalman Filtering, R. Brown, P. Hwang, Wiley Press, 1997.
- [ 4 ] T. Curtin, J. Bellingham, J. Catipovic, D. Webb, "Autonomous Oceanographic Sampling Network, Oceanography, Vol. 6, No. 3, pp.86-94, 1993.
- [ 5 ] M. Dhanak, E. An, K. Holappa, An AUV Survey in the Littoral Zone: Small-scale Subsurface Variability Accompanying Synoptic Observations of Surface Currents, IEEE Transactions of Oceanic Engineering, Vol.26, No.4, pp.752-768, October 2001.
- [ 6 ] Applied Optimal Estimation, A. Gelb, MIT Press, 1986.

- [ 7 ] G. Grenon, E. An, S. Smith, A. Healey, Enhancement of the Inertial Navigation System for the MORPHEUS Autonomous Underwater Vehicles, IEEE Transactions of Oceanic Engineering, Vol.26, No.4, pp.548-560, October 2001.
- [ 8 ] M. S. Grewal and A. P. Andrews, Kalman Filtering: Theory and Practice, Prentice Hall, 1993.
- [ 9 ] Einar Gustafson, Edgar An, Alexander Leonessa, Samuel Smith, A Post-Processing Kalman Smoother for Underwater Vehicle Navigation, UUST 2001, Trends New Hampshire, NH, September, 2001.
- [ 10 ] J. Huddle, in Inertial Systems Technology for High Accuracy AUV Navigation, AUV Navigation Workshop, pp.63-73, 1998, Boston, Mass.
- [ 11 ] R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, Journal of Basic Engineering, March 1960, pp. 35 - 46.
- [ 12 ] Smith, S.M., An, P.E., Holappa, K., Whitney, J., Burns, A., Nelson, K., Heatzig, E., Kempfe, O., Henderson, E., Kronen, D., Pantelakis, T., Font, G., Dunn, R., Dunn, S.E. Morpheus: ultra modular AUV for coastal survey and reconnaissance. *IEEE Journal of Oceanic Engineering*, 2001.
- [ 13 ] S.M. Smith, K. Ganesan, P.E. An, S.E. Dunn, Strategies for Simultaneous Multiple AUV Operation and Control, International Journal of Systems Science, Vol.29, No.10, pp.1045-1063, 1998.
- [ 14 ] T.R. Stockton et. al., Acoustic Position Measurement: An Overview, Offshore Technology Conference, Dallas, Texas, 1975.

## **2.9 Chemical Sensors**

**PI: Dr. Richard Granata**

### **Task 3.29**

#### **2.9.1 Summary**

This section describes the formulation of a chemical method to detect underwater trace explosives, as well as the design of a field-deployable device to implement the chemical method. The research goals are identified, the primary test materials, equipment and experiments are described and the results are discussed. The chemical compound, europium thenoyltrifluoroacetone, has been identified as an integral part of a viable underwater chemical detection method for underwater explosive traces. Included in this section is the final report on the enhanced capabilities (chemical species and sensitivity) of a chemical sensor UUV payload for detection of explosive materials for UUV applications.

#### **2.9.2 Introduction**

The ultimate purpose of this UUV component is to detect underwater explosives and provide a signal so that action can be taken. This process breaks down to three basic steps: (1) Obtain an underwater sample for testing, (2) Analyze the obtained sample and (3) Provide feedback of the results so that appropriate action can be taken.

Several methods exist to analyze a water sample for explosive traces [1], but practicality in UUV application dictates several limitations, such as size, cost, autonomy and processing speed. Consequently, these limitations in conjunction with the unique seawater environment eliminate most existing explosive detection methods. The research contained herein focuses on the formulation and testing of a detection method based on fluorescent tagging and the development of a field-deployable device to detect waterborne explosive traces with this method. Attention has been given to UUV parameters such as size, cost, power consumption, autonomy, analysis speed and sensitivity.

The research goals have been identified as follows:

- Evaluate the feasibility of developing a photoluminescent method of detecting underwater explosive traces.
- Examine different fluorescent compounds, looking for optimal combinations of the chosen fluorescer (europium) and sensitizing ligands to achieve both fluorescent loss in water (quenching) and maintained fluorescence in response to explosive compounds. Different combinations, concentrations and mixing orders of the chemicals are evaluated. Other factors that influence the

performance of the compounds are also evaluated, such as the amount of solvent required and/or used to deliver the chemicals into the seawater solution.

- Characterize the excitation and emission frequencies of the sensitized compounds.
- Evaluate the hypothesis that europium complexes will preferentially bond with explosive compounds over water molecules in an aqueous environment. Continue this experiment to include a seawater environment.
- Identify, purchase and set up a working underwater explosive detection device and develop a test plan. This goal includes concentration studies.
- Explain some potential fluorescence quenching issues of seawater that may cause the detection method to perform differently in seawater than in distilled fresh water.

### **2.9.3 Methods, Assumptions, and Procedures**

#### **2.9.3.1 Primary Test Materials**

The primary test materials include the explosive sample and the chemicals involved. Medical nitroglycerin (NG) tablets are used for the explosive sample to accommodate safety issues of the university. The method should be extendable to a wide range of explosive compounds based upon nitro chemistries. Europium is used for the fluorescer and two compounds were evaluated as sensitizing ligands: Thenoyltrifluoroacetone (TTA), ( $C_8H_5F_3O_2S$ ) and 1,10 Phenanthroline Monohydrate (OP), ( $C_{12}H_8N_2 \bullet H_2O$ ).

#### **2.9.3.2 Primary Test Equipment**

For the first stage of laboratory testing, the primary test equipment consisted of a handheld UV light (approximately 370 nm) and a Perkin-Elmer LS50B luminescence spectrometer. The handheld UV light was used to execute preliminary evaluations of different chemical mixtures under different conditions. This provided a quick, efficient method of testing the design path, without performing tedious, exact experiments for all possibilities. The luminescence spectrometer was used to precisely evaluate certain mixtures for fluorescence and quenching. The luminescence spectrometer can either record the light output of a compound with a given excitation wavelength, or it can scan for the best excitation wavelength to produce the maximum intensity of a given emission wavelength.

For the second stage of testing, the focus is on a working field detector design. The core of the design is a compact, underwater fluorometer (Figure 2.9.1). The fluorometer used in this experiment is a WETStar model, made by Wet Labs, Inc., that has been specially modified to provide 370 nm excitation and record 613 nm emission. Due to the low velocity, laminar flow that is fed into the fluorometer, an in-line static mixer is also utilized to assure proper mixing of the seawater and reagent solutions.



**Figure 2.9.1:** WETStar Fluorometer.

### **2.9.3.3 Experiments**

The luminescence spectrometer was used to evaluate the appropriate excitation and emission wavelengths for the chosen fluorescent compounds. Special attention was given in determining the excitation wavelength so that it corresponded to a standard, commercially available, LED light source. This consideration was included so that an LED light source could be used in the field-deployable device. Background fluorescence analyses were conducted for several solutions to provide additional insight into the real fluorescence change between explosive-laden and explosive-absent solutions. Several fluorescent compounds were compared to determine the best choice of sensitizing ligands, concentrations and mixing orders. The effect of the solvent that was used to deliver the chemicals into the solution was evaluated for its effect on the explosive detecting ability. The detection limit of nitroglycerin in the luminescence spectrometer of the chosen compound was evaluated. The performance difference between seawater and fresh water was evaluated to determine if the additional constituents of seawater affect the detection method. The customized LED spectrometer was ordered and its performance evaluated.

### **2.9.4 Results and Discussion**

The following section is taken from the completed thesis conclusions section [2] which summarizes the finding of this study.

It was determined that the use of a lanthanide element to fluorescently tag explosive traces is a viable underwater trace explosive detection method. While water quenches europium compound fluorescence, water-borne nitroglycerin is able to protect europium's fluorescent properties. This likely occurs because the explosive trace's negatively charged nitrite moiety is more strongly attracted to the positively charged lanthanide ion's free bonding site than dipolar water molecules are.

To capture the fluorescent properties of a lanthanide ion, radiation-absorbent ligands must be attached to absorb and transfer energy to it. The type of ligand is important, as well as mixing order if multiple ligands are used. It was found that the europium / thenoyltrifluoroacetone (Eu/TTA) complex produced significantly better results in underwater explosive detection than europium / thenoyltrifluoroacetone / 1,10-phenanthroline (Eu/TTA/OP) and europium / 1,10-phenanthroline / thenoyltrifluoroacetone (Eu/OP/TTA) complexes. Eu/TTA fluoresced strongly in the presence of NG, but almost completely lost fluorescence when NG was absent. On the other hand, Eu/OP/TTA and Eu/TTA/OP fluoresced strongly with and without water-borne NG. This suggests that the OP ligand creates a hydrophobic environment around the europium ion, even when NG is not present. The presence of the OP ligand also significantly reduced the solubility of the compound in methanol. Additionally, while Eu/TTA/OP and Eu/OP/TTA solutions contained the same ratios of components, they performed differently, indicating the importance of ligand mixing order.

It was found that the excitation wavelength required to create fluorescence of a lanthanide compound depended strictly on the excitation wavelengths of the attached ligands. When the TTA ligand was used, optimal excitation was found to be 382 nm and when the OP ligand was added, strong excitation also occurred around 310 nm. Excitation near the TTA requirement is easily accomplished via LED sources, whereas the deep ultraviolet wavelengths required by OP are not. Because of this and the better explosive-detection performance without OP, OP was omitted to provide an optimum compound for use. Since the thesis is ultimately aimed at a working design, practicality was factored in and excitation was chosen to be 370 nm for experimentation, versus the optimum wavelength of 382 nm. This choice was made because 370 nm is a standard wavelength available in LED's. To verify the correctness of this choice, testing was conducted on the Eu/TTA compound with both 370 nm and 382 nm excitation wavelengths for comparison, which indicated that very little performance is lost by this shift in excitation. Even less loss is expected in the field due to the fact that the 370 nm and 382 nm gap is closed somewhat due to the actual width of each one's excitation peak.

It is sometimes possible for the characteristic emission wavelength of an element to shift when it is combined with other components to form a compound. It was found that the characteristic europium emission wavelength of 613 nm persisted, regardless of the compound configuration. This wavelength did not change in the presence of OP, TTA, nitroglycerin or sodium, or in fresh water and seawater solutions.

Because europium fluorescence is quenched by water, it was necessary to combine the europium and sensitizing ligands in methanol before introduction into the seawater and water solutions. It was found that the methanol affects both the final solution clarity and fluorescence. Overall, the less methanol included, the better. For the tests conducted with Eu/TTA, fluorescence fell to negligible levels when the methanol level reached 35 percent of the total solution. Only Eu/TTA was tested for methanol effect because it was chosen as the more favorable compound in an earlier test. Rough solubility limits of the compounds were ascertained to provide some insight into the minimum amount of methanol required. OP had a negative effect on solubility. The maximum solubilities found for Eu/TTA, Eu/TTA/OP and Eu/OP/TTA were  $1.02 \times 10^{-2}$  M, were  $4.57 \times 10^{-3}$  M and were  $4.53 \times 10^{-3}$  M, respectively.

The europium detection method was found to perform considerably better in fresh water than in seawater. A specified amount of nitroglycerin could be detected in fresh water with less than 1/12 the amount of reagent required to detect the same amount of nitroglycerin in seawater. Based on references [3-6], it is believed that this is due to metal-exchange reactions with calcium and magnesium in the seawater. References [3-6] also note that acidic conditions negatively affect europium compound fluorescence. The impact of metal-exchange reactions and low pH were not quantified because the calcium and magnesium content of seawater is not expected to vary significantly from the seawater samples used for experimentation and the range of seawater pH is much higher than the problem ranges reported in references [3-6]. However, tests were conducted to prove that this explosive detection method is susceptible to these conditions and help explain the difference in performance between seawater and freshwater. These tests confirmed that this detection method is compromised by large amounts of calcium ions and low pH.

With Eu/TTA at  $1 \times 10^{-4}$  M concentration (total solution), nitroglycerin could be detected in the laboratory luminescence spectrometer down to concentrations as dilute as approximately  $1 \times 10^{-6}$  M.

After characterizing the chemical detection method in the laboratory with a luminescence spectrometer, tests were performed with a modified commercial fluorometer to move towards a field-deployable design. Static (non-flowing) tests indicated that, with this chemical detection method, a deployable fluorometer is sensitive to nitroglycerin dissolved in seawater. The sensitivity depends on the amount of the europium complex used, with more Eu/TTA translating to better sensitivity. In the WETStar characterization tests, sensitivity was found to be  $2.44 \times 10^{-7}$  M nitroglycerin with the equipment used, a Eu/TTA concentration in methanol of  $4 \times 10^{-4}$  M, and a mixing ratio of 8 percent. This translates to about 28 ppb. However, there is a limit to which the Eu/TTA concentration can be increased before problems are encountered with the particular fluorometer used in this experiment (WETStar). If the Eu/TTA concentration is high enough that the upper output voltage limit (5 V) of the WETStar was surpassed, the WETStar output information that defied visual observation and luminescence spectrometer readings. At these high Eu/TTA concentrations, the WETStar indicated that there was less intense fluorescence with nitroglycerin than without, even though it was

visually obvious that the opposite was true. Based on these comparisons, it was concluded that the WETStar output was erroneous when the Eu/TTA concentration was too high. Therefore, the best performance with this method and equipment is attained when the Eu/TTA complex is as high as possible, without reaching the point where the fluorometer outputs false results (possible off-scale digital-analog conversion). While higher europium complex concentrations bring better sensitivity (before saturation), they also require more reaction time. Until the reaction is completed, the fluorescence output oscillates erratically and produces little usable information. All of the concentrations studied needed less than five minutes to stabilize. Reaction time must also be considered in system design.

The impact of sample filtration was also addressed, and it was found that filtration slightly increases the fluorescence intensity reading from the fluorometer. This slight increase was noted in both the nitroglycerin-laden and nitroglycerin-absent solutions, with very little change in their relative readings. With minimal change in fluorometer output and no noticeable change in relative readings, filtration adds little value to the design. However, if a pump is used to pass the sample through the fluorometer, a minimum amount of filtration will be required to assure pump operation and endurance.

The flow-through trace-explosive detector design was validated with a laboratory hydraulic system. This system combined the seawater/nitroglycerin solution with the europium complex solution in an appropriate ratio and then mixed them, before the final solution was passed through the modified WETStar fluorometer. Using this system, the fluorometer was able to discriminate between plain seawater and seawater that contained traces of nitroglycerin, and the design concept was proven.

It is believed that the negatively charged nitrite moiety of the nitroglycerin compound is what makes it detectable with the chemical method presented herein. Because this characteristic is common to many explosive types, it is believed to be highly likely that this method can be extended to detect many explosive types, in addition to nitroglycerin.

Based on this research, two proposed design options are shown below in Figures 2.9.2 and 2.9.3. The first design utilizes two small pumps, while the second makes use of one pump and a restrictor combination to control the seawater / reagent ratio. The UUV speed cannot be assumed to be constant, and because the mixing ratio of the seawater and reagent must be controlled, at least one pump is necessary. The two-pump design would be easier to setup, while some tuning would be required to achieve the proper mixing ratio with the restrictor setup. The restrictor setup would be less expensive and likely require less maintenance. Cursory research indicates that pumps and restrictors are available that meet the requirements of this design. For example, Micropump, Inc., can provide suitable pumps, and The Lee Company produces a range of hydraulic restrictor sizes that will fit this application. Many companies make small pumps, but this application is quite demanding for miniature pumps. The pumps must be accurate in their flow rates and more importantly; they must be able to withstand the internal case pressure that results from water depths that the CCST UUV must be designed to. Static mixers are available from a variety of companies. TAH Industries provided the static



mixer used in the proof of design test of this thesis. Work is proceeding with the two pump design.

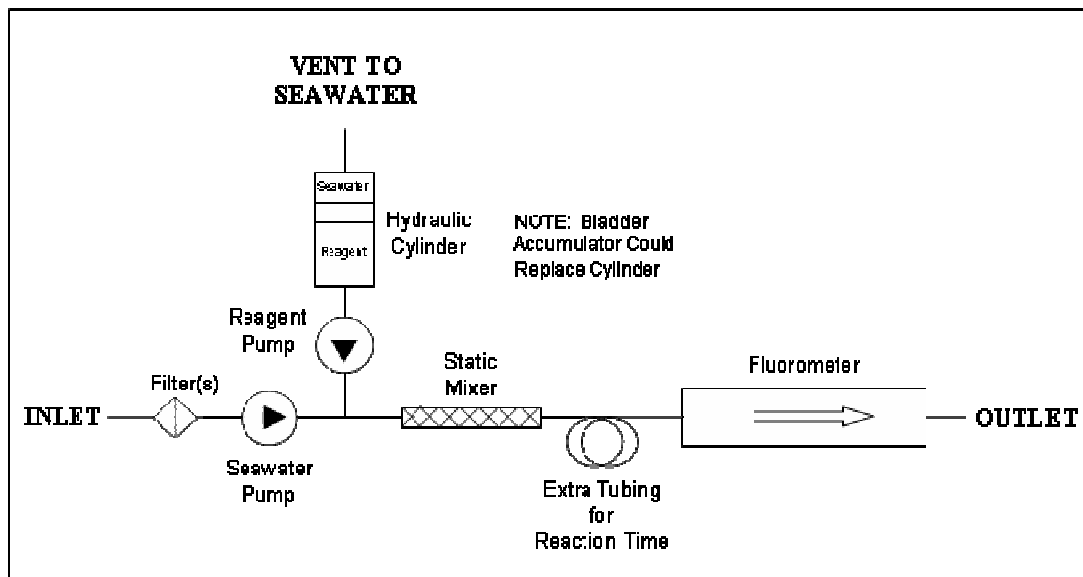


Figure 2.9.2: Proposed Design Schematic no. 1

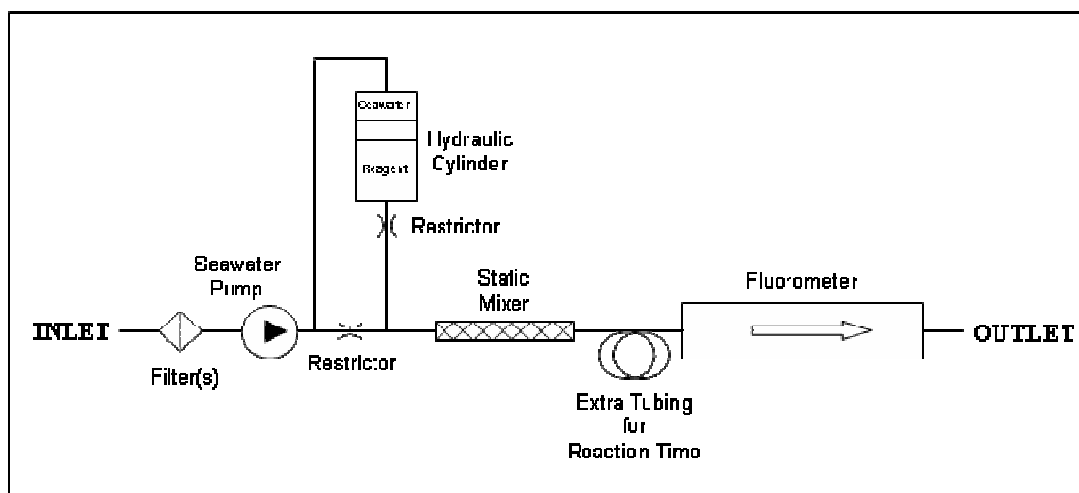


Figure 2.9.3: Proposed Design Schematic no. 2

## 2.9.5 Conclusions and Recommendations

From this work, it was determined that the use of a lanthanide element to fluorescently tag explosive traces is a viable underwater trace explosive detection method. Europium was used as the lanthanide element. While water quenches (shortens) the europium compound's fluorescence, water-borne nitroglycerin enhances (prolongs) its fluorescence.

To capture the fluorescent properties of a lanthanide ion, radiation absorbent ligands must be attached to absorb and transfer energy to it [7]. The type of ligand is important, as well as mixing order if multiple ligands are used. Thenoyltrifluoroacetone (TTA) is a good ligand to use with europium for this purpose. Ortho-phenanthroline (OP) is not recommended because its absorption range does not coincide with that attainable with LED light sources and it appears to prevent fluorescence quenching when explosive traces are not present.

The combination of europium and TTA is recommended as a compound to detect underwater explosive traces. It is also recommended to limit the amount of chemical solvent (methanol) to as low a percentage as practicable, definitely not to exceed 20%.

Experiments have been completed to allow selection of final chemical detection method. Also, a sensor module has been identified, modifications specified, the UUV operable hardware ordered and received. Work is now underway for the next phase of the work which is laboratory and field testing of the chemical sensor module installed within the UUV.

**Deliverable:** A final report on the enhanced capabilities (chemical species and sensitivity) of a chemical sensor UUV payload for detection of explosive materials for UUV applications – This section (2.9) and reference 2 (T.A. Langston, below).

## 2.9.6 References for Section 2.9 - Chemical Detector

- 1) J. Yinon, S. Zitrin, Modern Methods and Applications in Analysis of Explosives, John Wiley & Sons Ltd., 1993.
- 2) T.A. Langston, "Chemical Method and Device to Detect Underwater Trace Explosives via Photo-Luminescence," M.S. Thesis, Florida Atlantic University, December, 2006.
- 3) C. N. Shtykov, T. D. Smirnova, Y. V. Molchanova, Synergistic Effects in the Europium(III)-thenoyltrifluoroacetone-1,10-Phenanthroline System in Micelles of Block Copolymers of Nonionic Surfactants and Their Analytical Applications, Chernyshevsky State University, Saratov, Russia, January 2001.
- 4) A. Adeyiga, P. Harlow, L. Vallarino, and R. Leif, Advances in the development of lanthanide macrocyclic complexes as luminescent bio-markers, Department of Chemistry, Virginia Commonwealth University, 2006.
- 5) Perkin Elmer Life Sciences, Stability of the Wallace LANCETM Eu-chelates, LANCETM Time-Resolved Fluorescence Detection Application Note.
- 6) Cisbio International, New Europium Cryptates to Probe Molecular Interactions Using HTRF.
- 7) E. Menzel, K. Bouldin, R. Murdock, Trace Explosives Detection by Photoluminescence, TheScientificWorld JOURNAL (2004) 4, 55–66 ISSN 1537-744X; DOI 10.1100/tsw.2004.7.

### **3.0 High Definition High Frame Rate Color Camera for Surveillance**

**PI: Dr. William E. Glenn**

#### **Tasks 3.30-3.34**

#### **3.1 Summary**

The overall objective of this segment of the project is to develop a high definition, high frame rate color video camera system for surveillance. During the first year of the program a 3840x2160 30P (30 FPS progressive scan) super-high-definition color CMOS camera—the HDMAX camera—with variable frame rate and remotely controlled infrared filter changer was designed, fabricated, tested, and demonstrated. This camera gathers 50 times the amount of information in its field of view as do standard-resolution video cameras often used for surveillance purposes. A flash-memory-based solid-state device for recording large amounts of image data generated by the camera was also designed, fabricated, and tested. Field tests demonstrated that the camera's high resolution makes it possible to do electronic zoom on sections of an image without permanent loss of the remaining portions of the field of view, and the high frame rate allows the use of moving target indication, velocity measurement, and the observation of brief events that help classify targets of interest. During the year covered by this report two upgraded HDMAX camera systems were built for use in next year's program in the investigation of 3D imaging, and a prototype video compression system was built and tested. Providing in excess of a 10:1 compression ratio, this compression system, when combined with a solid-state recorder module, will allow nearly three hours of recording time per module. The HDMAX camera, video signal compressor, and solid-state recorder are ideally suited for video surveillance on ships, submarines, harbors, AUVs, and drone aircraft.

#### **3.2 Objective, Tasks, and Deliverables**

The objective of this segment of the project is to develop a high definition, high frame rate color video camera for surveillance. The proposed research, which includes the development of a memory unit for large amounts of imaging data generated by the camera and a video compression system to reduce that data by a factor of ten or more, will contribute to the improvement of imaging technology for use in surveillance.

Tasks specified for this year's program are the following:

1. Communicate and coordinate with the Navy to define the High Definition Maximum (HDMAX) camera system requirements.
2. Modify the 2004 HDMAX design and fabricate two upgraded HDMAX camera systems for subsequent use in 3D video experiments.
3. Modify the 2004 HDMAX display interface and fabricate two of the upgraded devices.
4. Add compression to the solid-state memory to allow increased record time.
5. Install HDMAX on an aircraft carrier and test the complete HDMAX system (HDMAX camera system includes HDMAX camera, display adaptor, display and

laptop computer) and complete two (2) additional field testes of the complete HDMAX camera system.

One deliverable item is specified for this segment of the program, a Sony 2160 Line Projector (CLIN 0007).

### 3.3 Results and Discussion

All tasks except #5, testing the camera system onboard an aircraft carrier, were completed. Jim Buss, at the time the ONR Program Officer overseeing this research program, had made preliminary arrangements for a field test of the HDMAX camera system on the USS Dwight D. Eisenhower. However, this past fall the carrier was ordered to sail to the Persian Gulf region thereby precluding the test. No substitute field test on an aircraft carrier has been arranged. The two remaining field tests were conducted last spring, one in New York City during Fleet Week and the other at Port Hueneme in California. We expect to receive guidance regarding any substitute field test from the current Program Officer.

Because the Sony 2160 Line Projector (CLIN 0007) is needed in connection with further investigations into 3D video image display under the Year 3 program, we have reached an agreement with the Program Manager to postpone its delivery until no later than 30 September 2007.

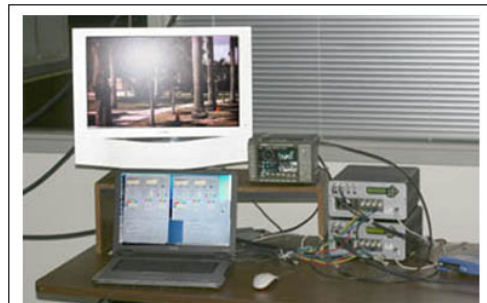
The largest-scale and most definitive field test of the camera was conducted last spring at Port Hueneme, California. The camera, along with four other video cameras brought together for testing by the Navy, was dock-mounted and used to image targets on the sea at distances up to eight miles. The results of this test have been detailed in a separate report and were discussed in last year's final report, though the tests were conducted in connection with this current year's effort. The camera, operating at 30 FPS progressive scan, could resolve the smallest detail of test charts mounted on a boat at 3 miles. At 5 miles, ground fog was the limiting factor, the natural scattering of the light reducing image color and contrast significantly. These tests indicated that, under typical port and ocean surveillance conditions at large distances and where light scattering is significant, it is preferable to use a monochrome HDMAX camera without an IR blocking filter. Penetration through the fog is improved, and higher sensitivity (and, therefore, ability to work at lower light levels) is also achieved.

Modification of the 2004 HDMAX design, fabrication of upgraded HDMAX camera systems, and modification and fabrication of two upgraded HDMAX display interfaces proceeded straightforwardly. Development of the video compression system, on the other hand, presented some challenges, subsequently overcome. Our original system concept was based on

JPEG 2000 IP (intellectual-property software) cores implanted in field-programmable gate arrays (FPGAs).



*Fig. 3.1. Pair of HDMAX cameras*



*Fig. 3.2. Two display interfaces with control computer.*

The cost of components and software for this approach was quite high, totaling \$130K. Fortunately, Analog Devices, Inc., recently announced the availability of an improved and low-cost JPEG 2000 video compression chip, the ADV212 JPEG 2000 Video Codec, and we are now compressing the raw video signal using this device, feeding the output of the device directly into our solid-state recorder unit. We note in passing that the ADV212 JPEG 2000 Video Codec, which has replaced the problematic ADV202, is poorly documented, and incorporating it in the system has been difficult. Current activity centers on operating all components of the system—camera, compressor, solid-state recorder, decompressor, and display—in tandem. During Year 3 of the program two such systems will be used to record and display 3D video scenes.

We note in passing that we have had many discussions with sensor design people at Panavision Imaging that have resulted in changes in design of that company's new Quad-HD RGB sensor array, which operates with 12 million pixels but at 120 FPS (progressive). The sensor, though produced, has yet to be tested by Panavision. Since Panavision could not determine if the sensor worked we could not use it in the present program as we had planned.

### **3.4 Patents Filed during this period:**

None of the patents filed during this period relate to this project.

### **3.5 Conclusion**

The HDMAX camera and solid-state recorder are ideally suited for video surveillance of harbors and on ships, submarines, AUVs, and drone aircraft. The camera gathers 50 times the amount of information in its field of view as do standard-resolution video cameras. The high resolution makes it possible to do electronic zoom on sections of the image without losing subsequent access to the full-field-of-view, high-resolution image. This aspect can be especially important for surveillance. The high frame rate allows the use of moving target indication, velocity measurement, and the observation of brief events that help classify targets of interest. The camera can operate as a color camera when color provides an important signature. Through removal of the IR blocking filter the camera is made more sensitive and better able to penetrate fog and haze. In this configuration it is preferable to use a monochrome sensor.

### **3.6 Recommendations**

In its current form the camera still needs some improvement, summarized here.

- Better temperature control.
- Software refinements to reduce the minor differences in gain applied to the eight sectors of the image.
- Automatic control of gain and iris diaphragm.
- Fully remote operation.
- Weatherproof and hardened case

In addition, the following additional features would give the camera additional value in surveillance situations:

- Person-operated electronic zoom of regions of interest
- Automatic detection of possible regions of interest
- Solid-state-memory–based instant replay
- 3D stereo from moving single camera

In connection with this last point, we note that the camera has important application to 3D imaging and measurement. Under a grant from NASA, for example, we demonstrated that a video camera of this kind can be used in the three-dimensional inspection of the heat-shield tiles on the Space Shuttle. With such an application in mind, the camera components have been partially space-qualified.

## 4.0 Stereo and Multi-View Image And Video Stabilization, Calibration, Coding, Analysis And Playback

PI: Dr. Borko Furht

### 4.1 Summary

This report reviews the second year of research activities in the field of image and video analysis algorithms for coastline security. Our research work in the second year has been focused on developing robust techniques and methodologies for multi-view video capturing, analysis, delivery and presentation. This work extends our efforts from the first year which mainly focused on developing algorithms and techniques for motion detection, object tracking, and object classification in maritime scenes from single-view images and video sequences. As a set of deliverables of the second year research we proposed and implemented robust algorithms for compensation of camera vibration, 3D reconstruction from multiple images, 3D video player for playback, algorithms for multi-view and 3D video compression and image and video object segmentation algorithms using depth information. Figure 4.1 illustrates how these research efforts fit in the overall project.

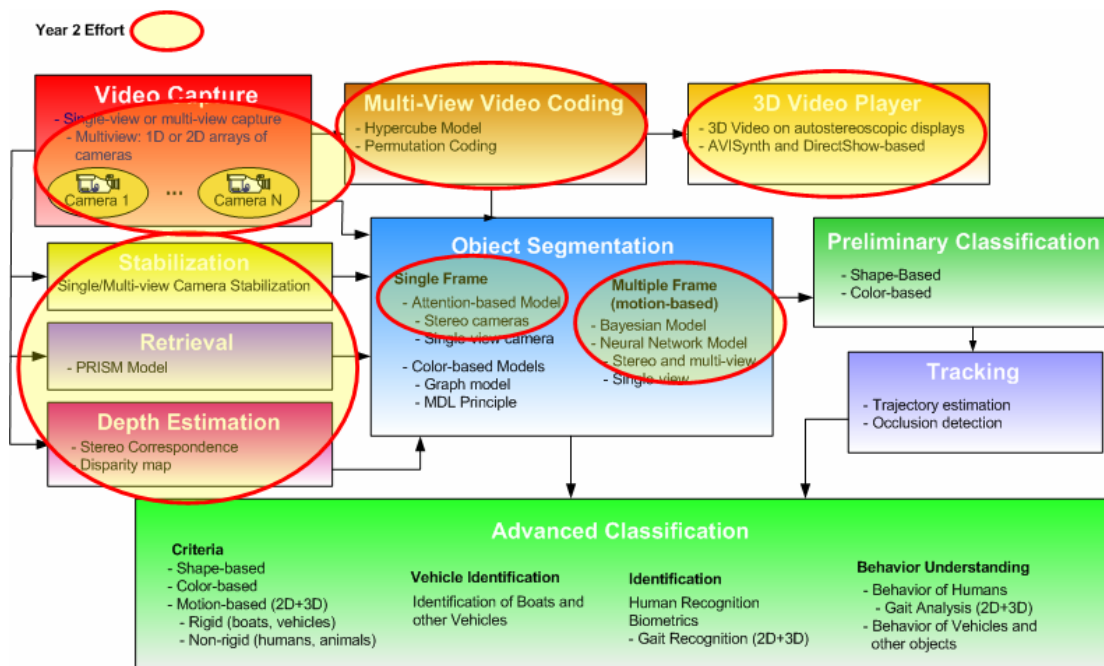


Figure 4.1 Overview of our research activities in the second year of the project

## 4.2 Introduction

### 4.2.1 Project Description

This report describes the work involved in the development of a video surveillance system for the Center of Coastline Security Technology. The ultimate goal of our work is to provide semi-automatic tools to monitor marine traffic at key locations, analyzing the contents of incoming video streams, detecting potential threats, and triggering the corresponding action.

Our efforts during Year 2 of the grant have been focused mostly on algorithms and techniques for robust camera stabilization and calibration, depth enhanced object segmentation and efficient delivery and presentation of multi-view videos. As it can be seen from Figure 4.1, these methodologies represent an essential task in video processing and form the foundation of scene understanding, object tracking, behavior analysis as well as the identification of humans, boats and other objects of interest, which will be addressed in the Year 3 research efforts. The developed methods are ultimately targeted for real-time or near-real-time processing from sequences obtained by both regular cameras as well as high-definition cameras supporting HDTV and/or QuadHDTV resolutions.

### 4.2.2 Project Scope and Objectives

This project is part of the Center of Coastline Security Technology at Florida Atlantic University. It is expected to be integrated at the output of the video capture stage developed by Dr. Bill Glenn's group. The objectives for Year 2 are:

- *Explore methods for video stabilization in order to remove undesirable motion effects so that only intentional motions effects are retained.* Causes of unwanted motion effects in video are usually atmospheric disturbances of surveillance cameras mounted on static poles or platforms, or unwanted motions of cameras in videos taken by hand or from mobile platforms. It is imperative to have methodologies for video stabilization since both aforementioned types of undesirable motions could be present in surveillance applications.
- *Investigate effective approaches to camera calibration and stereo correspondence.* For camera calibration, both intrinsic and extrinsic parameters of a camera system must be determined. To estimate depth from stereo or multi-view captured images and videos, a corresponding disparity map must be computed.
- *Explore effective approaches to 3D video compression, delivery and playback.* Techniques and methods for efficient compression of stereo and multi-view sequences is an ongoing research area. It is anticipated that 3D video improves surveillance applications. 3D autostereoscopic displays (no glasses required) are recently being released and are becoming notably inexpensive. The goal is also to create a 3D video player for Sharp autostereoscopic display, which is one of the first commercially available autostereoscopic displays.



- *Investigate methods and algorithms for image and video object segmentation using depth information.* Single-view object segmentation is limited in the sense that the occlusion is difficult to detect and it is difficult to distinguish far objects from close ones. Object segmentation with depth information allows for easy occlusion detection, allows distinguishing far from close objects and helps classification and tracking.

#### 4.2.3 Project Team

Faculty:

Dr. Borko Furht, PI

Dr. Taghi M. Khoshgoftaar, co-PI

Dr. Oge Marques

Dr. Hari Kalva

Dr. Daniel Socek

Students:

Lakis Cristodoulou, Alvaro Fonseca, Qiming Luo, Liam Mayron, Carlos Pertuz, and Xiaoyuan Su

### 4.3 An Empirical Study on Video Stabilization

#### 4.3.1 Problem Statement

In recent years digital videos have grown rapidly, due to the dramatic cost reduction and performance improvement of devices for acquiring and processing videos, such as cameras and computers. However, videos taken by hand or from mobile platforms often suffer from undesirable motion effects, which are caused by the unwanted motions of cameras. In addition, surveillance cameras mounted on static poles or platforms are also subject to atmospheric disturbances. As a result, the visual quality of collected videos is degraded. The objective of video stabilization, also known as image sequence stabilization (ISS), is to remove undesirable motion effects so that only intentional motion effects are retained. The primary benefit of video stabilization is improving video quality, and in the context of surveillance applications, resulting in better performance measured by receiver operating characteristics (ROC) [1]. In addition, video stabilization has a desirable side effect of reducing the bit rate for encoding the stabilized videos [2].

#### 4.3.2 Literature Survey

AA In the past 20 years, many approaches on video stabilization have been proposed, and they can be divided into three categories:

1) Mechanical:

Mechanical devices such as accelerometers, gyros, and mechanical dampers [3] are employed to reduce sensor platform vibrations of relatively large magnitudes.

**2) Optical:**

An optical system is designed and implemented to compensate for unwanted camera motion using motion sensors and active optical system. An example is the Nikon Camera VR (Vibration Reduction) System.

**3) Electronic (software based):**

Stabilization is implemented as a post-processing operation after videos are collected. This is in contrast to the previous two categories, where stabilization is performed during video capturing. The typical strategy is to estimate the inter-frame motions and then to filter out unwanted motions while preserving intentional motions.

The focus of this study is on software based stabilization techniques. Many algorithms have been proposed in this area, and they vary in terms of accuracy, computational speed and memory requirements. They usually operate in two stages: motion estimation and motion correction. The motion estimation stage is critical since the accuracy of estimated motion parameters ultimately determines the effectiveness of the stabilization process.

The motion estimation stage aims to estimate the global motions between frames, and existing approaches can be classified into the following categories:

**1) Phase correlation:**

Based on the translation property of the Fourier transform, the cross-power spectrum of two images related by pure translation is an impulse, which indicates the parameters of translation. Furthermore, if two images are related by a combination of translation, rotation, and scaling, those parameters can be inferred from their Fourier magnitude spectra in polar representations

[4]. The advantages of this approach for handling translational jitters include: low computational cost, insensitivity to illumination changes, graceful performance degradation with non-pure translation [5].

**2) Block matching:**

[6] applies full search block matching to binary version of input images to estimate block motion vectors. After outliers are rejected at both local and global levels, least mean squares is employed to estimate the affine motion parameters.

**3) Optical flow:**

[7] applies a modified Lucas and Kanade's method to estimate the optical flow for non-overlapping blocks, and then uses trimmed least squares to estimate the affine motion parameters. [8] estimates the affine motion parameters by minimizing a robust cost function defined on all pixels. [9] estimates an optical flow field through local cross-correlation analysis. The estimation is implemented in a multi-resolution coarse-to-fine fashion on Laplacian pyramid images. Then an affine motion model is fit using weighted least-squares regression. [10] applies EM algorithm to simultaneously estimate affine motion models for multiple objects, which enable selective stabilization of foreground, background, or a combination of objects.

4) Feature tracking:

[11] tracks features between frames using a multi-resolution coarse-to-fine scheme in a Laplacian pyramid. The tracked features are used to fit a 2D affine motion model. [12] applies Kalman filtering to track a set of features extracted by Lucas and Kanade's method. A robust rejection rule X84 is employed to identify outliers, then global motion parameters are estimated using least squares.

5) Integral projection:

[13] estimates the horizontal and vertical motions of a single large block centered on each frame, reducing a 2D matching problem to a 1D matching problem, thus resulting in a dramatic saving in computational cost. [14] applies integral projection matching to the entire image. Projection are obtained from a sub-sampled set of rows and columns to reduce computational cost. [15] proposes to use one of the three color components in RGB images for integral projection matching.

6) Grid points:

[1] estimates the translation vector by minimizing a correlation index computed on a selected set of points on a fixed grid, overlaid on both the reference image and the current image. A point is selected if it has been recently tracked.

After the global motions between consecutive frames are obtained from the motion estimation stage, they can be accumulated in order to compute the motion parameters of each frame with respect to a reference frame. The motion parameters reflect not only intentional camera motions, but also unwanted motions. Intentional motions are usually smoother and more regular than unwanted motions. Therefore, in the frequency spectrum of signals of motion parameters, intentional motions consist of low frequency components, while unwanted motions consist of high frequency components.

The general strategy of motion correction is to apply low pass filtering in order to filter out the unwanted motions. [16] compares motion vector integration (MVI) and frame position smoothing (FPS). MVI applies a first order low pass IIR filter to the parameter signal. It can be implemented on-line for real time processing. FPS is implemented either by DFT domain filtering or by time-delayed IIR filtering. [6] applies MVI with an adaptive damping coefficient to better cope with the trade-off between removing unwanted motions against preserving intentional motions. In the case of translational jitter, [17] shows that Kalman filtering performs better than MVI in terms of retaining image content and preserving intentional motions. [8] applies Kalman filtering to estimate intentional motion parameters, assuming an affine motion model. [10] applies Kalman filtering to estimate affine motion parameters in both the forward and backward directions to overcome the delay in forward only filtering.

### 4.3.3 Evaluation of Motion Estimation Methods

We have implemented the following motion estimation methods and evaluated their performance: phase correlation [4], KLT-based feature tracking, KLT-based block

matching, and integral projection [14]. Among the four algorithms, the first two are capable of dealing with motions involving a combination of translation, rotation and scaling, while the last two are suitable for pure translations.

#### **4.3.3.1 Motion Models**

In the context of motion estimation for video stabilization, the motion between video frames are assumed to follow one of the two-dimensional projective linear transformations listed below.

##### **4.3.3.1.1 Translation**

This transformation consists of translation only, represented by the parameters  $d_x$  and  $d_y$ .

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (1)$$

##### **4.3.3.1.2 Euclidean Transformation**

This transformation consists of a combination of translation and rotation. The rotation is parameterized by the angle  $\theta$ .

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & d_x \\ \sin(\theta) & \cos(\theta) & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (2)$$

##### **4.3.3.1.3 Similarity Transformation**

This transformation consists of a combination of translation, rotation, and isotropic scaling. The scaling is parameterized by the factor  $s$ .

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s \cos(\theta) & -s \sin(\theta) & d_x \\ s \sin(\theta) & s \cos(\theta) & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3)$$

An equivalent form is:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \end{pmatrix} \begin{pmatrix} s \cos(\theta) \\ s \sin(\theta) \\ d_x \\ d_y \end{pmatrix} \quad (4)$$

In order to determine the motion parameters of the similarity transformation between two frames based on a set of matched feature points with locations  $(x_i, y_i) : i = 1, \dots, n$  and

$(x'_i, y'_i) : i = 1, \dots, n$ , a system of linear equations can be solved by singular value decomposition.

$$\begin{pmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \vdots \\ \vdots \\ x'_n \\ y'_n \end{pmatrix} = \begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{pmatrix} \begin{pmatrix} s \cos(\theta) \\ s \sin(\theta) \\ d_x \\ d_y \end{pmatrix} \quad (5)$$

#### 4.3.3.1.4 Affine Transformation

This transformation consists of a combination of translation, rotation, and non-isotropic scaling.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (6)$$

An equivalent form is:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{pmatrix} \quad (7)$$

In order to determine the motion parameters of the affine transformation between two frames based on a set of matched feature points with locations  $(x_i, y_i) : i = 1, \dots, n$  and  $(x'_i, y'_i) : i = 1, \dots, n$ , a system of linear equations can be solved by singular value decomposition.

$$\begin{pmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \vdots \\ \vdots \\ x'_n \\ y'_n \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ x_n & y_n & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_n & y_n & 1 \end{pmatrix} \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{pmatrix} \quad (8)$$

Morimoto and Chellappa [18] conducted a comparative study on the performance of different motion models in image sequence stabilization. The models include Euclidean, similarity, and affine transformations. They concluded that more complex models perform worse than simple models due to their sensitivity to tracking errors.

#### 4.3.3.2 Integral Projection

The idea of integral projection is to convert the problem of matching two-dimensional blocks into the problem of matching one dimensional vectors, thus dramatically reducing the computational cost. One type of vector is obtained by summing pixels along each row. The corresponding vectors in two frames are matched to determine the vertical movement between the two frames. Similarly, the other type of vector is obtained by summing pixels along each column. The corresponding vectors in two frames are matched to determine the horizontal movement between the two frames. This approach relies on a strong assumption that motion in the direction perpendicular to the estimation direction is small. This approach is attractive because of its low computational cost. However, in our tests, the accuracy is not satisfactory except for simple synthetic images.

#### 4.3.3.3 Phase Correlation

Phase correlation is based on the Fourier shift theorem [4]. If two images  $f_1$  and  $f_2$  differ only by a translation vector  $(d_x, d_y)$ , as described by Equation 9, then their corresponding Fourier transforms  $F_1$  and  $F_2$  satisfy Equation 10.

$$f_2(x, y) = f_1(x - d_x, y - d_y) \quad (9)$$

$$\frac{F_1(u, v)F_2^*(u, v)}{|F_1(u, v)F_2(u, v)|} = e^{j2\pi(ud_x + vd_y)} \quad (10)$$

The left side of Equation 10 is known as the cross-power spectrum, and its inverse Fourier transform is an impulse, with the peak located at the translation vector  $(d_x, d_y)$ .

If two images  $f_1$  and  $f_2$  are related by not only translation, but also a rotation with angle  $\theta$ ,

$$f_2(x, y) = f_1(x \cos \theta + y \sin \theta - d_x, -x \sin \theta + y \cos \theta - d_y) \quad (11)$$

then the magnitude of  $F1$  and  $F2$  satisfy:

$$M_1(\rho, \phi) = M_2(\rho, \phi - \theta) \quad (12)$$

where  $(\rho, \phi)$  are the polar coordinates in the Fourier domain. Therefore,  $\theta$  can be determined by phase correlation.

If two images  $f1$  and  $f2$  are related by scaling with factors  $(a, b)$ ,  $F1$  and  $F2$  satisfy

$$F_2(u, v) = \frac{1}{|ab|} F_1(u/a, v/b) \quad (13)$$

$$G_2(\log u, \log v) = \frac{1}{|ab|} G_1(\log u - \log a, \log v - \log b) \quad (14)$$

where  $F(u, v) = F(e^{\log u}, e^{\log v}) = G(\log u, \log v)$ . Therefore,  $(a, b)$  can also be determined by phase correlation on logarithmic scale.

In summary, if two images are related by a combination of translation, rotation, and scaling, the motion parameters can be determined using phase correlation based on Equations 10, 12, and 14.

#### 4.3.3.4 KLT-based feature tracking

The Kanade-Lucas-Tomasi algorithm [19,20,21] is one of the most reliable techniques for estimating optical flow, based on an empirical performance study by Barron et al. [22]. Consider the task of tracking a feature window  $W$  in two adjacent frames of an image sequence  $I(x, y, t)$ . If the sampling interval  $\tau$  is small enough, it is acceptable to assume that the feature window undergoes a translation represented by a displacement

vector  $\mathbf{d} = (d_x, d_y)^T$ :

$$I(x, y, t) = I(x + d_x, y + d_y, t + \tau) \quad (15)$$

To determine  $\mathbf{d}$ , the sum of squared difference (SSD) residual  $\epsilon$  in the feature windows is minimized.

$$\epsilon = \sum_{(x,y) \in W} [I(x + d_x, y + d_y, t + \tau) - I(x, y, t)]^2 \quad (16)$$

When the displacement vector is small,  $I(x + d_x, y + d_y, t + \tau)$  can be approximated by its first-order Taylor expansion:

$$I(x + d_x, y + d_y, t + \tau) \approx I(x, y, t) + \frac{\partial I}{\partial x} d_x + \frac{\partial I}{\partial y} d_y + \frac{\partial I}{\partial t} \tau \quad (17)$$

or in matrix form,

$$I(x + d_x, y + d_y, t + \tau) \approx I(x, y, t) + \mathbf{g}^T \mathbf{d} + I_t \tau \quad (18)$$

where

$$\mathbf{g} = \begin{pmatrix} I_x \\ I_y \end{pmatrix} = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{pmatrix}$$

and

$$I_t = \frac{\partial I}{\partial t}$$

Substitute Equation 18 into Equation 16,

$$\epsilon \approx \sum_{(x,y) \in W} (\mathbf{g}^T \mathbf{d} + I_t \tau)^2 \quad (19)$$

To solve for the displacement  $\mathbf{d}$  that minimizes  $\epsilon$ , differentiate  $\epsilon$  with respect to  $\mathbf{d}$  and set the derivative to zero. Then we get the following equation:

$$\mathbf{Z} \mathbf{d} = \mathbf{e} \quad (20)$$

where

$$\mathbf{Z} = \sum_{(x,y) \in W} \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

and

$$\mathbf{e} = -\tau \sum_{(x,y) \in W} I_t \begin{pmatrix} I_x \\ I_y \end{pmatrix}$$

The feature window  $W$  can be tracked only if Equation 20 can be reliably solved, which requires that the two eigenvalues of  $\mathbf{Z}$  be larger than the noise level. When both eigenvalues are small, the feature window  $W$  contains roughly constant intensity levels. A large eigenvalue and a small eigenvalue indicate a unidirectional texture pattern. Two large eigenvalues can represent corners or any other pattern that can be tracked reliably.

In practice, the eigenvalues are bounded from above due to the limited range of pixel values. A feature window  $W$  is trackable if the minimum of the two eigenvalues exceeds a predefined threshold.

#### 4.3.3.5 Handling Outliers in Motion Estimation by X84 Rule

The goal of motion estimation is to determine the motion of the background, which results from the movement of the camera. However, in real scenes, there often exists moving foreground objects. When tracked features happen to appear on foreground objects, the estimated motion parameters also reflect the motions of foreground objects.



Therefore, it is important to reduce the impact of those outlier features so that the estimated motion corresponds to the motion of the camera only. We apply a robust and efficient outlier detection method called X84 rule [23], as proposed in [12]. The advantage of X84 rule in comparison to other outlier detection approaches such as LMedS [24] or RANSAC [25] is that it takes much less computational cost, which is crucial for real-time video stabilization and analysis.

In this study, we choose the similarity transformation to model the inter-frame motion because of the following two reasons:

- 1) As stated in subsection 4.3.3.1, more complex models perform worse than simple models due to their sensitivity to tracking errors.
- 2) The motion parameters in Equation 3 (scale factor, rotation angle, horizontal and vertical translations) have straightforward meanings, in comparison to those in Equation 6.

In practice, if affine transformation is required to model complex motion, the procedure for handling outliers is similar.

Let  $\hat{s}, \hat{\theta}, \hat{d}_x, \hat{d}_y$  be the estimated motion parameters of the similarity transformations, the estimation errors for the features are defined by

$$e_i = \sqrt{(x'_i - \hat{x}'_i)^2 + (y'_i - \hat{y}'_i)^2} \quad (21)$$

where  $\hat{x}'_i$  and  $\hat{y}'_i$  represent the estimated location of feature  $i$  in the current frame, while  $x'_i$  and  $y'_i$  represent the actually measured location of feature  $i$  in the current frame.  $\hat{x}_i$  and  $\hat{y}_i$  are obtained by

$$\begin{pmatrix} \hat{x}'_i \\ \hat{y}'_i \\ 1 \end{pmatrix} = \begin{pmatrix} \hat{s} \cos(\hat{\theta}) & -\hat{s} \sin(\hat{\theta}) & \hat{d}_x \\ \hat{s} \sin(\hat{\theta}) & \hat{s} \cos(\hat{\theta}) & \hat{d}_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix}, 1 \leq i \leq n \quad (22)$$

where  $x_i$  and  $y_i$  represent the location of feature  $i$  in the previous frame.

The set of estimation errors  $\{e_i : i = 1, \dots, n\}$  form a distribution, and X84 rule defines feature  $k$  as an outlier if its error differs from the median of errors by more than 5.24 times the mean absolute deviation (MAD):

$$e_k - m > 5.24 \times \text{MAD} \quad (23)$$

where

$$m = \text{median}\{e_i : 1 \leq i \leq n\}$$

and

$$\text{MAD} = \text{median}\{|e_i - m| : 1 \leq i \leq n\}$$

Equation 5 is first applied to obtain an initial estimate of the motion parameters. Next the X84 rule is applied to identify outliers from all detected features. Then Equation 5 is applied for the second time on all the features except the outliers to obtain the final estimate of the motion parameters.

#### **4.3.3.6 KLT-based block matching**

This approach considers the whole frame as a feature window and applies KLT-based feature tracking to match this macro window. In practice, the feature windows is a sub-image of a frame, obtained by chopping off four border stripes whose widths correspond to the maximum possible translation between adjacent frames. The advantage of this approach in comparison to traditional block matching is that it is a hierarchical estimation algorithm and it is capable of providing sub-pixel accuracy.

#### **4.3.3.7 Test Results using Pairs of Images with Synthetic Motion**

We summarize the test results on the four methods as follows:

1) Tests on pairs of images with synthetic motion involving translation only

Integral projection is not capable of generating reliable estimation of motion parameters (horizontal and vertical translations) for our test images

KLT-based block matching is capable of generating estimates with sub-pixel accuracy with reasonable computation cost. On a PC with Intel 2.8GHz CPU, it takes 0.09 second to estimate the motion between two frames of size  $300 \times 200$ .

The phase correlation approach is only capable of providing an estimate rounded to the nearest integer. This is because, in the original formulation of the algorithm, the parameters are estimated by finding the location of a discrete impulse on the image grid. Although interpolation techniques can be applied to provide better accuracy, it still produces inferior results in comparison to KLT-based feature tracking and KLT-based block matching.

2) Tests on pairs of images with synthetic motion involving a combination of translation, rotation, and scaling.

The computation time of phase correlation makes it infeasible for real-time processing. On a PC with Intel 2.8GHz CPU, it takes 1.64 second to estimate the motion between two frames of size  $300 \times 200$ .

On a PC with Intel 2.8GHz CPU, it takes about 0.09 second (or 11 frames per second) for the KLT-based feature tracking algorithm to estimate the motion between two frames of size  $300 \times 200$ .

Therefore, KLT-based feature tracking algorithm is the only algorithm capable of dealing with vibrations involving a combination of translation, rotation, and scaling in real time.

#### 4.3.3.8 Generating Synthetic Vibrations

In this study, since real surveillance videos with vibrations are currently not available, we generate synthetic videos to test the performance of motion estimation methods. The synthetic videos are produced by imposing synthetic vibrations into each frame of real surveillance videos.

The surveillance videos in our study are provided by the Nova Southeastern University Oceanographic Center Waterway Expert Traffic System Project. The videos were taken on the side of an urban canal in Fort Lauderdale in 1998. The camera was fixed on the canal bank, and only the side views of the passing ships were captured in the videos. The video format is MPEG-1 with a resolution of 352×240 and 24-bit color depth.

Given a video sequence, the first frame is used as the reference frame. The transformation matrix between the reference frame and all other frames can be obtained by accumulating the transformations between consecutive frames:

$$\mathbf{T}_n^0 = \mathbf{T}_1^0 \mathbf{T}_2^1 \mathbf{T}_3^2 \dots \mathbf{T}_n^{n-1} \quad (24)$$

It is assumed that the motion parameters of vibrations follow harmonic models, which is commonly used to characterize mechanical vibrations. The following seven types of vibrations are based on a combination of translation, rotation, and scaling, where

$\mathbf{T}_n^0[k]$  represents the transformation matrix between frame n and the reference frame for vibration type k.

$$\mathbf{T}_n^0[1] = \begin{pmatrix} 1 & 0 & d_x(n) \\ 0 & 1 & d_y(n) \\ 0 & 0 & 1 \end{pmatrix} \quad (25)$$

$$\mathbf{T}_n^0[2] = \begin{pmatrix} \cos(\theta(n)) & -\sin(\theta(n)) & 0 \\ \sin(\theta(n)) & \cos(\theta(n)) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (26)$$

$$\mathbf{T}_n^0[3] = \begin{pmatrix} \cos(\theta(n)) & -\sin(\theta(n)) & d_x(n) \\ \sin(\theta(n)) & \cos(\theta(n)) & d_y(n) \\ 0 & 0 & 1 \end{pmatrix} \quad (27)$$

$$\mathbf{T}_n^0[4] = \begin{pmatrix} s(n) & 0 & 0 \\ 0 & s(n) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (28)$$

$$\mathbf{T}_n^0[5] = \begin{pmatrix} s(n) & 0 & d_x(n) \\ 0 & s(n) & d_y(n) \\ 0 & 0 & 1 \end{pmatrix} \quad (29)$$

$$\mathbf{T}_n^0[6] = \begin{pmatrix} s(n) \cos(\theta(n)) & -s(n) \sin(\theta(n)) & 0 \\ s(n) \sin(\theta(n)) & s(n) \cos(\theta(n)) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (30)$$

$$\mathbf{T}_n^0[7] = \begin{pmatrix} s(n) \cos(\theta(n)) & -s(n) \sin(\theta(n)) & d_x(n) \\ s(n) \sin(\theta(n)) & s(n) \cos(\theta(n)) & d_y(n) \\ 0 & 0 & 1 \end{pmatrix} \quad (31)$$

The motion parameters are defined by:

$$d_x(n) = 10 \sin(\omega n) \quad (32)$$

$$d_y(n) = 10 \sin(\omega n) \quad (33)$$

$$\theta(n) = 10 \sin(\omega n) \quad (34)$$

$$s(n) = 1 + 0.1 * \sin(\omega n) \quad (35)$$

where

$$\omega = \frac{2\pi}{250} \quad (36)$$

and they are shown in Figure 4.2. The motion parameters follow the same harmonic model in terms of phase and frequency, to enable the performance evaluation of stabilization algorithms at the worst case when all the parameters reach extreme values simultaneously.

#### 4.3.3.9 Measuring the Accuracy of Motion Estimation Methods

In previous work [18], the accuracy of motion estimation methods is evaluated by a measure called peak signal-to-noise ratio (PSNR) defined below:

$$PSNR(I_1, I_0) = 10 \log \frac{255^2}{MSE(I_1, I_0)} \quad (37)$$

where

$$MSE(I_1, I_0) = \frac{1}{wh} \sum_{x=1}^h \sum_{y=1}^w (I_1(x, y) - I_0(x, y))^2 \quad (38)$$

where  $w$  and  $h$  are the width and height of the image, respectively. PSNR measures the divergence between the stabilized image and desired stabilization result, which can be due to a variety of reasons, such as noise, errors of estimated motion parameters, distortion due to inaccurate motion models, interpolation errors, etc. Based on PSNR, two measures of stabilization can be computed. Inter-frame transformation fidelity (ITF) is defined as the PSNR between two consecutive stabilized frames, while global transformation fidelity (GTF) is defined as the PSNR between the current stabilized frames and the reference frame.

The measures based on PSNR do not require knowledge of the ground truth of the motion parameters. However, this measure has several shortcomings. First, PSNR does not consider non-overlapping regions, which can be a serious problem in the computation of GTF when the current frame has no overlap with the reference frame due to large motion. In this case, a lower bound has to be specified to detect such a scenario [18]. Second, PSNR is only an indirect measure of the accuracy of estimated motion parameters, because the MSE formula (Equation 38) is not only dependent on the deviation of the estimated motion parameters, but also dependent on the spatial distribution of pixel values. When an image contains a large amount of pixels with high local gradient, even small errors in estimated motion parameters could cause large values of MSE. Conversely, when an image contains only a small amount of pixels with high local gradient, large errors in estimated motion parameters may not translate into large values of MSE.

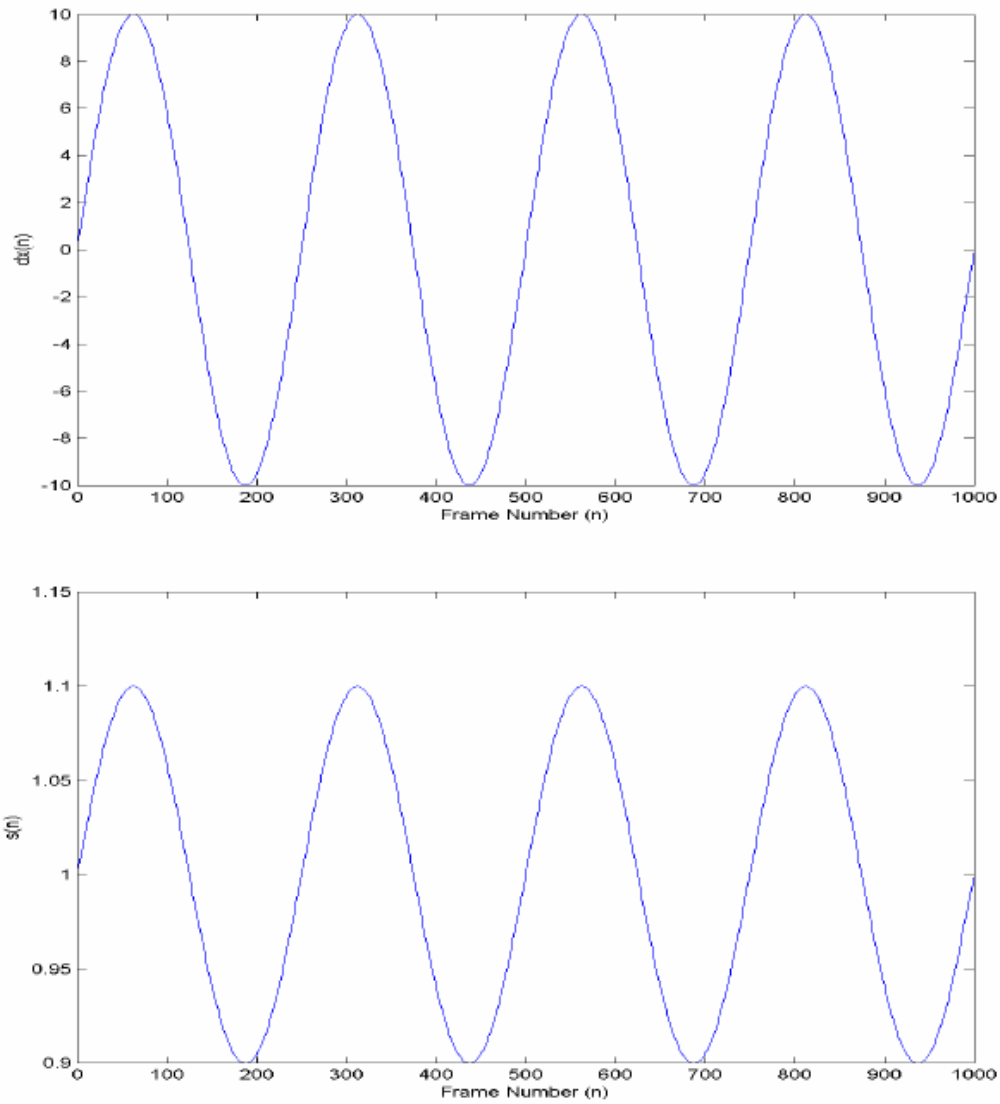


Figure 4.2 Vibrations modeled by motion parameters

Since the vibrations are generated synthetically, the true motion parameters are readily available. We propose a measure called “average pixel deviation” (APD) to directly assess the accuracy of estimated motion parameters in comparison to true motion parameters.

Given a video sequence, the first frame is used as the reference frame. The transformation matrix between the reference frame and all other frames can be obtained by accumulating the transformations between consecutive frames:

$$\mathbf{T}_n^0 = \mathbf{T}_1^0 \mathbf{T}_2^1 \mathbf{T}_3^2 \dots \mathbf{T}_n^{n-1} \quad (39)$$

This computation can be carried out based on both true motion parameters as well as estimated motion parameters. Let

$$\mathbf{T}_n^0 = \begin{pmatrix} s \cos(\theta) & -s \sin(\theta) & d_x \\ s \sin(\theta) & s \cos(\theta) & d_y \\ 0 & 0 & 1 \end{pmatrix} \quad (40)$$

and

$$\hat{\mathbf{T}}_n^0 = \begin{pmatrix} \hat{s} \cos(\hat{\theta}) & -\hat{s} \sin(\hat{\theta}) & \hat{d}_x \\ \hat{s} \sin(\hat{\theta}) & \hat{s} \cos(\hat{\theta}) & \hat{d}_y \\ 0 & 0 & 1 \end{pmatrix} \quad (41)$$

denote the true transformation matrix and the estimated transformation matrix, respectively. Then APD for frame n is defined by

$$APD_n = \sqrt{\sum_{x=1}^h \sum_{y=1}^w [(x' - \hat{x}')^2 + (y' - \hat{y}')^2]} \quad (42)$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s \cos(\theta) & -s \sin(\theta) & d_x \\ s \sin(\theta) & s \cos(\theta) & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (43)$$

$$\begin{pmatrix} \hat{x}' \\ \hat{y}' \\ 1 \end{pmatrix} = \begin{pmatrix} \hat{s} \cos(\hat{\theta}) & -\hat{s} \sin(\hat{\theta}) & \hat{d}_x \\ \hat{s} \sin(\hat{\theta}) & \hat{s} \cos(\hat{\theta}) & \hat{d}_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (44)$$

where x and y represent the location of a pixel in the reference frame; x' and y' represent the corresponding location of the pixel in the current frame after applying the true transformation matrix;  $\hat{x}'$  and  $\hat{y}'$  represent the corresponding location of the pixel in the current frame after applying the estimated transformation matrix; w and h are the width and height of the frame.

#### 4.3.3.10 Test Results using Videos with Synthetic Motion

In this study, the surveillance videos were taken on the side of an urban canal, where the traffic is sparse. Most of the time there are no ships in the scene. In order to test the robustness of the motion estimation algorithm against outliers, we have selected 16 video segments with different types of significant foreground motion. Each video segment contains 1000 frames. For each video segment, we impose the seven types of vibrations to each frame according to equations 25 to 31, and generate seven synthetic videos. Then we apply the KLT-based feature tracking algorithm to compute the transformation matrix between the reference frame and all other frames by accumulating the transformations between consecutive frames. The computed motion parameters are compared with the ground truth to obtain APD measure. In Figures 4.3 and 4.4, we show two key frames of two of the video segment as well as the APD measure under the seven types of vibrations with respect to frame number.

From the figures, we can observe the following:

- 1) APD is not correlated to the amount of translation.

- 2) APD is highly correlated to the magnitudes of the rotation angle and the scaling factor. This is because the KLT-based feature tracking algorithm is based on the assumption that the motion of features are translational only, and its performance starts to degrade when the rotation angle and the scaling factor becomes larger.
- 3) In most of the figures, the largest APD value is about 7, which coincides with the largest magnitudes of the rotation angle and the scaling factor.
- 4) Rotation is the dominating cause of large values of APD, which is evident by comparing the peak values of APD in sub-figures d, e, h, and i against c, f, and g. Scaling is a moderate contributor to large values of APD. The effect of translation is small, and remains almost constant within the tested range, despite the variation of the amount of translation.



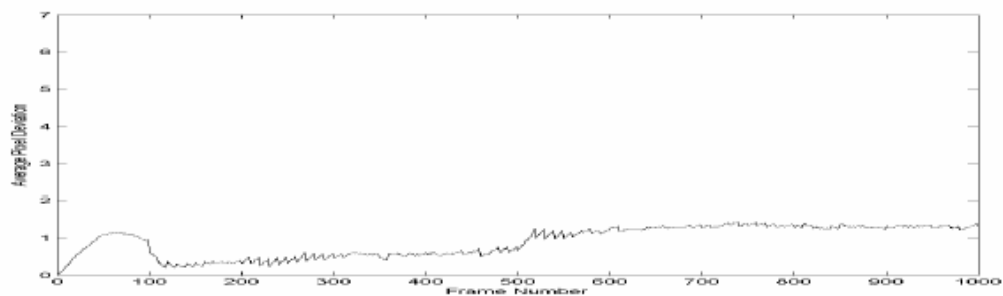
## Center for Coastline Security Technology Year Two-Final Report



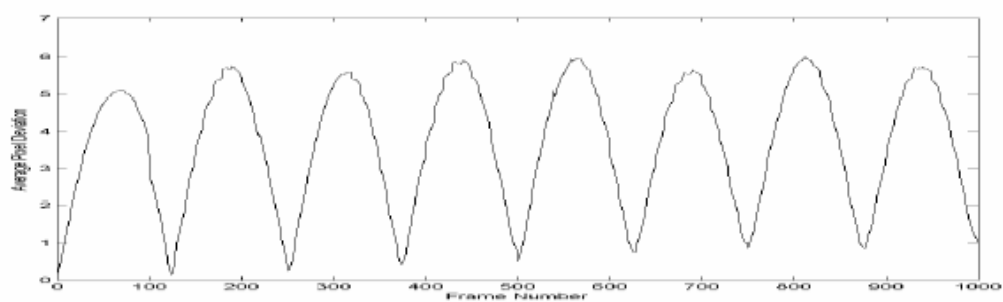
(a)



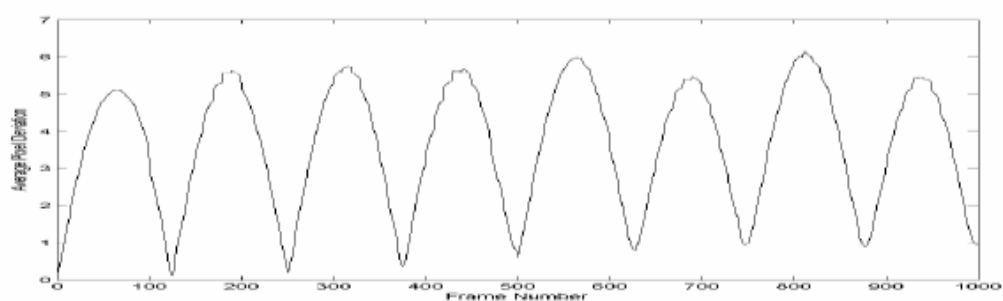
(b)



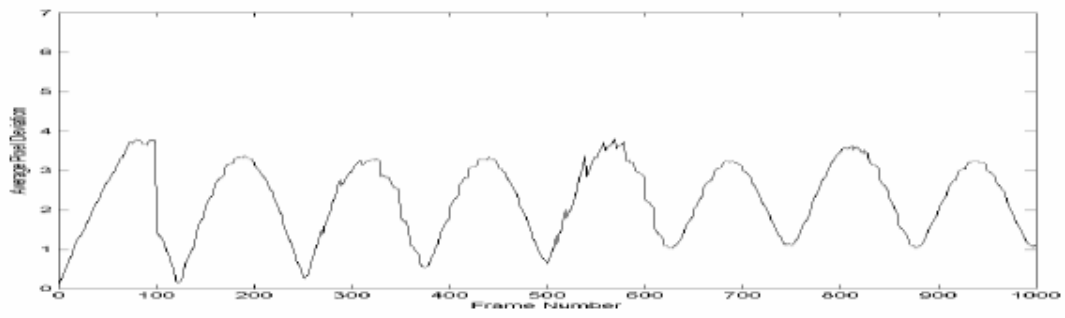
(c) Translation



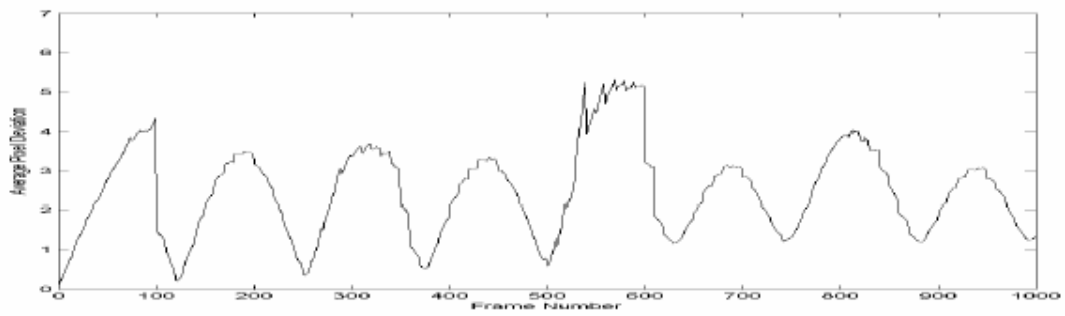
(d) Rotation



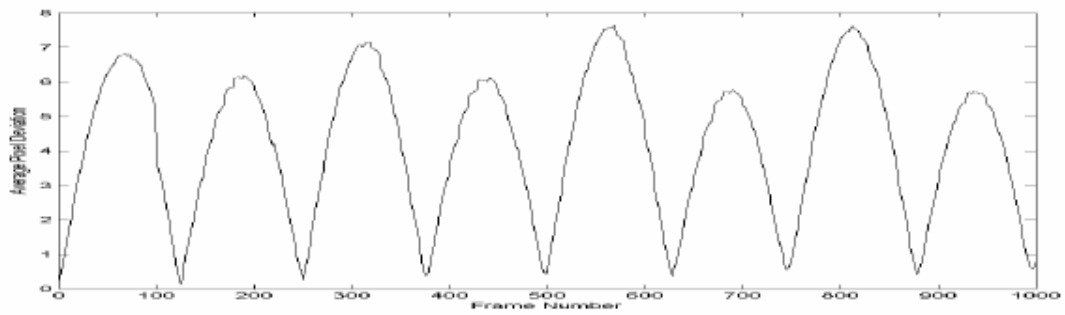
(e) Translation and Rotation



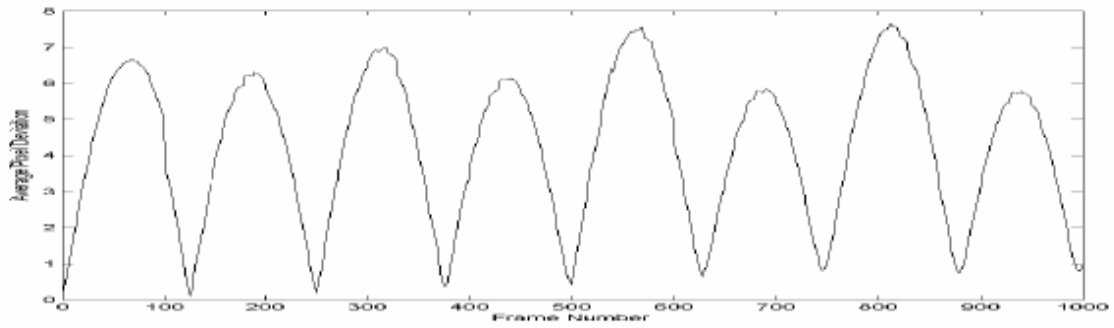
(f) Scaling



(g) Translation and Scaling



(h) Rotation and Scaling



(i) Translation, Rotation and Scaling

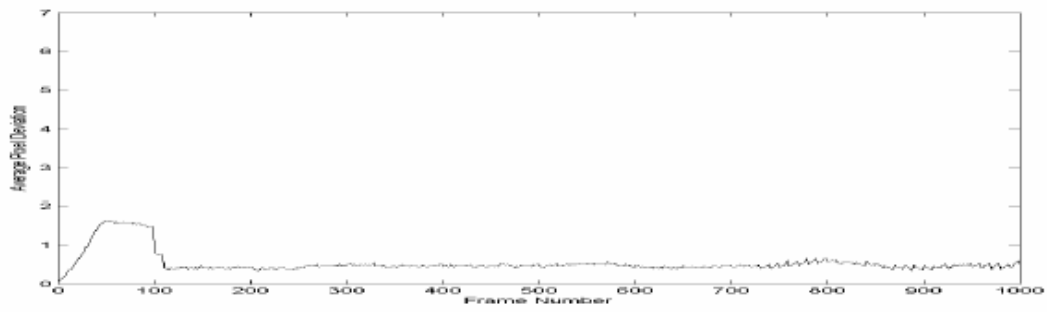
Figure 4.3 Test video sequence I and corresponding motion diagrams



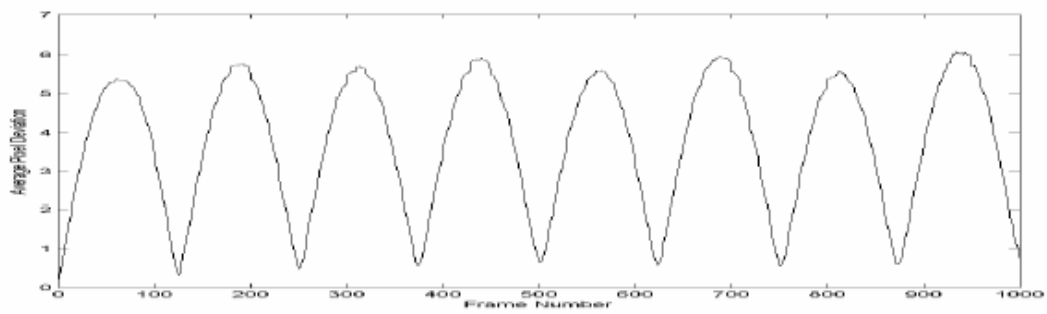
(a)



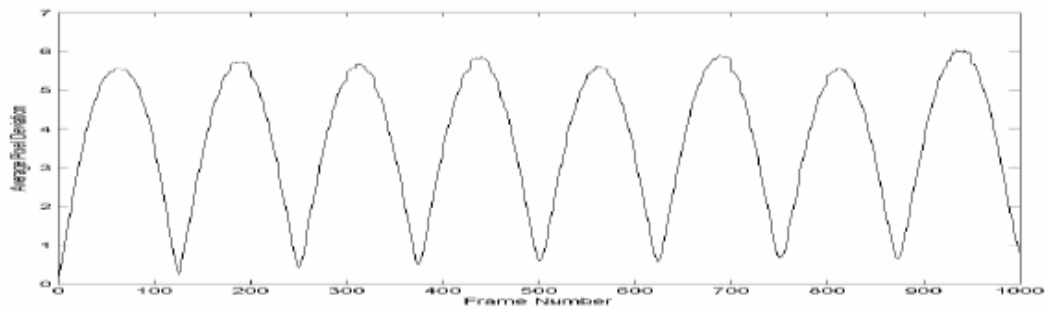
(b)



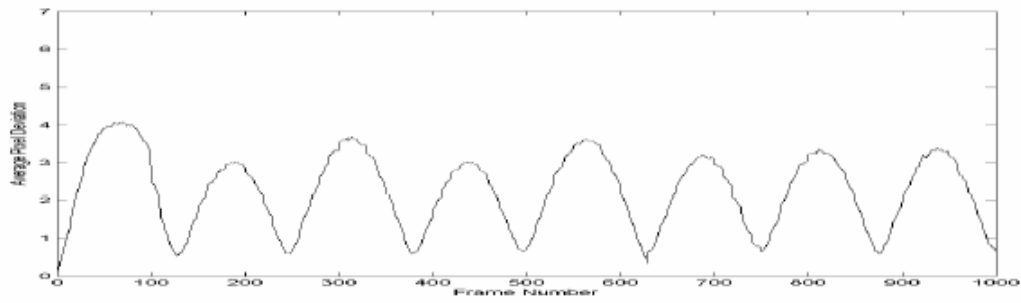
(c) Translation



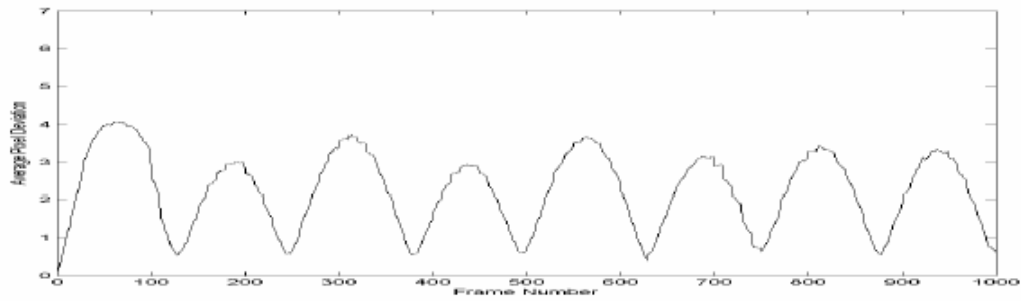
(d) Rotation



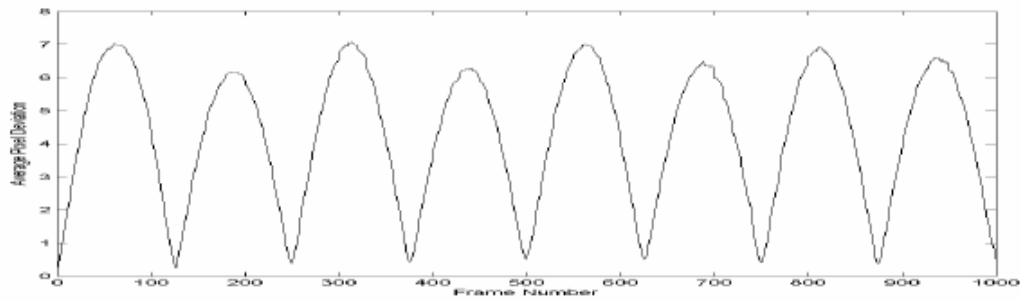
(e) Translation and Rotation



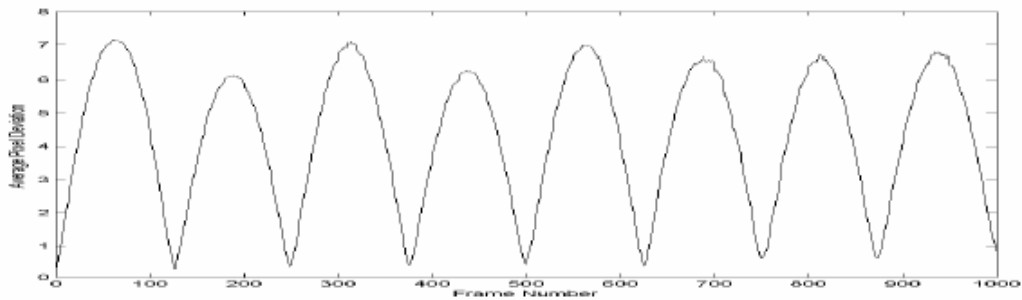
(f) Scaling



(g) Translation and Scaling



(h) Rotation and Scaling



(i) Translation, Rotation and Scaling

Figure 4.4 Test video sequence II and corresponding motion diagrams

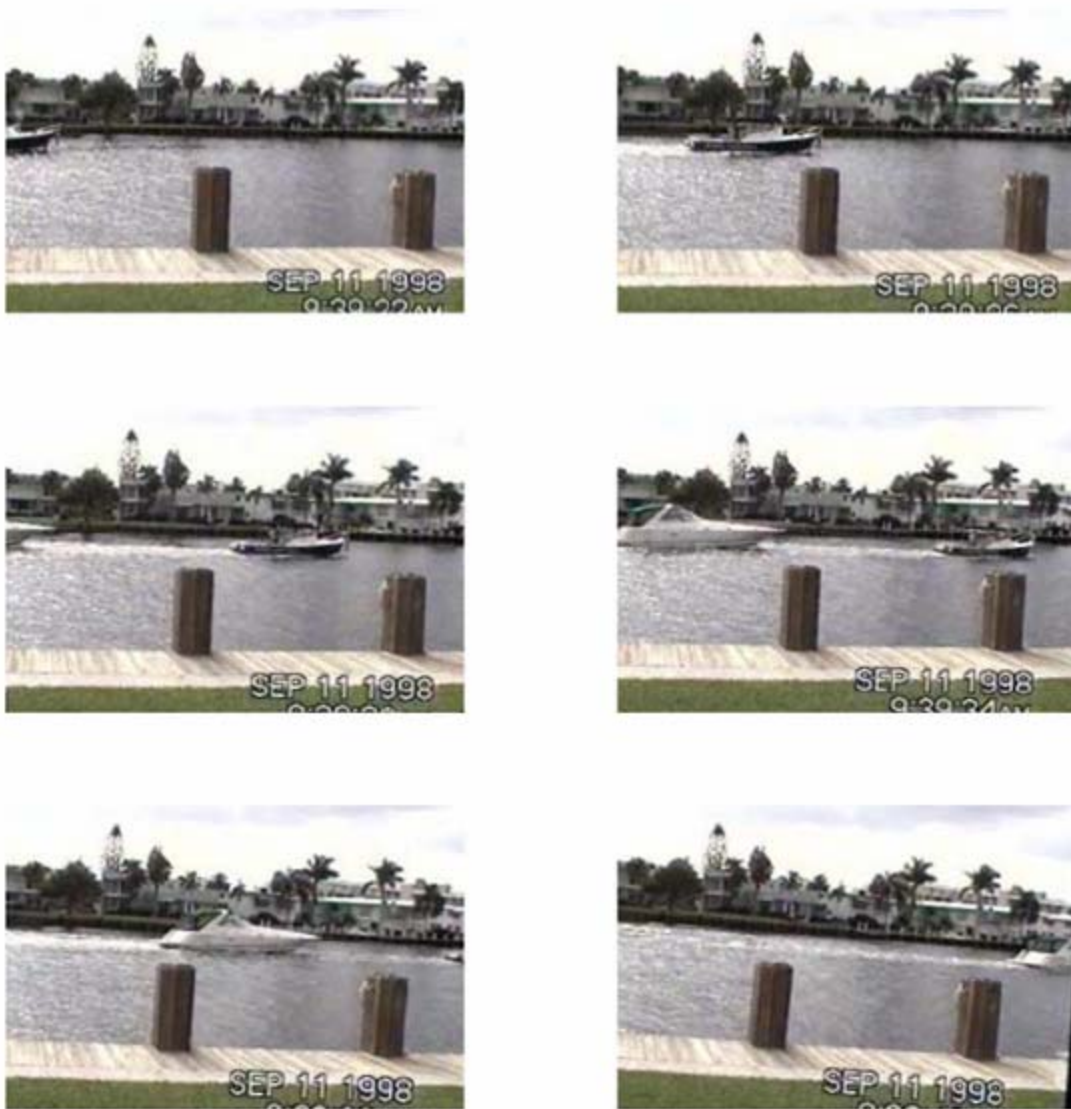


Figure 4.5 Test videos and corresponding APD values (left to right, top to down): 2.83, 6.08, 9.64, 13.83, 20.29, and 26.29.

#### 4.3.3.11 Reducing Error Accumulation

When applying equation 39 to compute the transformation matrix between the current frame and the reference frame, a practical issue of error accumulation often arises, which has not been addressed in previous studies to the best of our knowledge. In this study, we propose a periodic correction method to reduce error accumulation.

In Figure 4.5, we compare the APD curve of the stabilization results on the 16 videos in the case of vibration type 7 (a combination of translation, rotation, and scaling), in the cases of with and without applying the proposed correction method. The error measured by APD grows to unacceptable levels (defined by a threshold of 10) in 9 of the 16 test videos. In four of the videos, APD grows to more than 20.

In our experience, when the maximum value of APD does not exceed 7, the stabilized frames do not appear jittery. When APD is greater than 10, the quality of stabilization becomes degraded.

Figure 4.5 shows examples of APD values of 2.83, 6.08, 9.64, 13.83, 20.29, and 26.29. In the first two images, the caption in the lower right corner is in the upright position. In contrast, they show gradually more apparent skew in the last four images when APD values grow from 9.64 to 26.29.

The proposed periodic correction method to reduce error accumulation is described as follows. For current frame  $n$ , we maintain a dynamic buffer of consecutive frames, with frame numbers between  $n-b_1$  to  $n-b_2$ .  $b_1$  and  $b_2$  are parameters that can be tuned for specific application domains to account for the different magnitudes of vibrations. For current frame  $n$ , one frame  $m$  is selected from the buffer that is closest to frame  $n$  in terms of the relative APD (RAPD) defined as follows:

$$RAPD(m, n) = \sum_{x=1}^h \sum_{y=1}^w [(x'_m - x'_n)^2 + (y'_m - y'_n)^2] \quad (45)$$

$$m = \arg \min_k \{RAPD(k, n), \quad n - b_1 \leq k \leq n - b_2\} \quad (46)$$

where

$$\begin{pmatrix} x'_m \\ y'_m \\ 1 \end{pmatrix} = \mathbf{T}_0^m \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (47)$$

$$\begin{pmatrix} x'_n \\ y'_n \\ 1 \end{pmatrix} = \mathbf{T}_n^0 \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (48)$$

Then we compute the transformation matrix between frame  $n$  and the reference frame by the following formula:

$$\mathbf{T}_n^0 = \mathbf{T}_m^0 \mathbf{T}_n^m \quad (49)$$

where  $\mathbf{T}_m^0$  is available at frame  $m$ , and  $\mathbf{T}_n^m$  can be obtained by applying the KLT-based feature tracking algorithm to frame  $m$  and frame  $n$ .

To reduce computational cost, RAPD can be computed by Equation 50, which is equivalent to Equation 45 by straightforward manipulation.

$$\begin{aligned}
 RAPD = & \sum_{x=1}^h x^2 [d_{11}^2 + d_{21}^2] + \sum_{y=1}^w y^2 [d_{12}^2 + d_{22}^2] + \\
 & 2 \sum_{x=1}^h \sum_{y=1}^w xy [d_{11} * d_{12} + d_{21} * d_{22}] + \\
 & 2 \sum_{x=1}^h x [d_{11} * d_{13} + d_{21} * d_{23}] + \\
 & 2 \sum_{y=1}^w y [d_{12} * d_{13} + d_{22} * d_{23}] + \\
 & w * h * [d_{13} * d_{13} + d_{23} * d_{23}]
 \end{aligned} \tag{50}$$

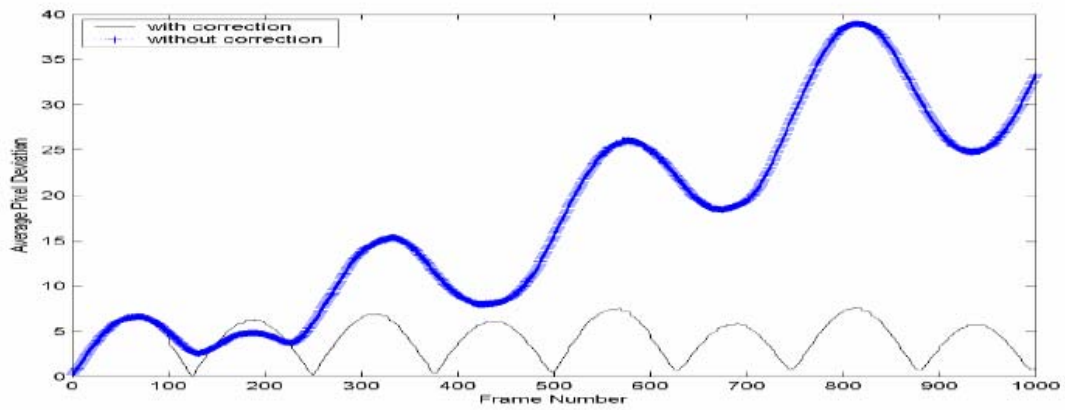
where

$$d_{ij} = a_{ij} - b_{ij}, \quad 1 \leq i, j \leq 3 \tag{51}$$

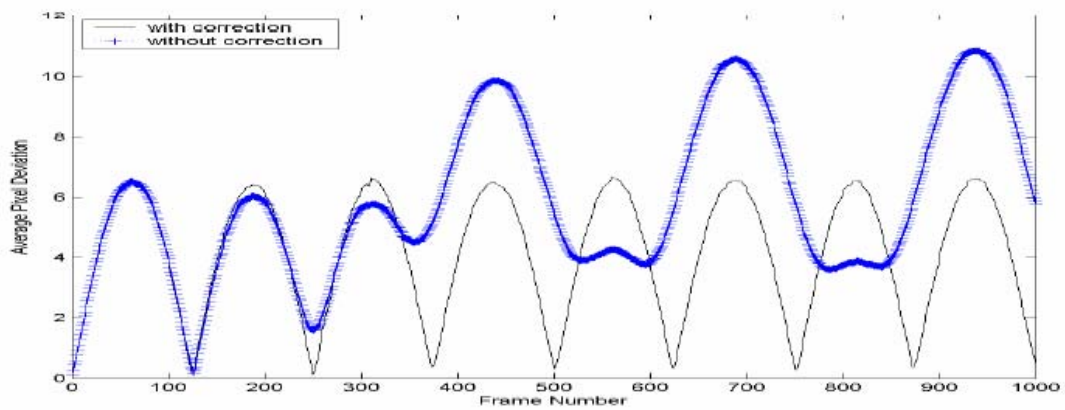
Note that all the sum terms are only dependent on the width and the height of frames, so they can be pre-computed off-line. Equation 50 reduces the computation cost of RAPD from  $O(wh)$  to  $O(1)$ .

This correction method does not need to be applied for each new incoming frame. Instead, it can be applied periodically for every  $T$  frames. The parameters  $b_1$ ,  $b_2$ , and  $T$  can be empirically determined to achieve a balance between the effectiveness of reducing error accumulation and reducing computational cost. In this study, the following parameter values are used:  $b_1 = 90$ ,  $b_2 = 10$ ,  $T = 10$ .

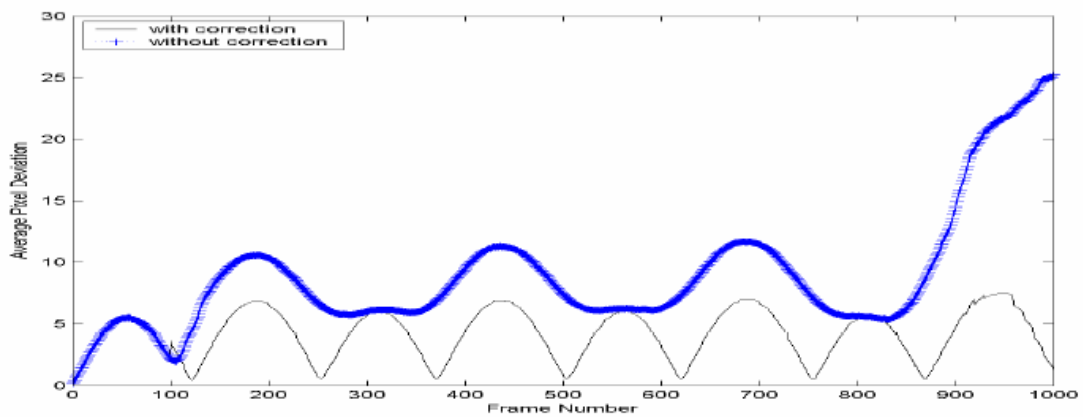




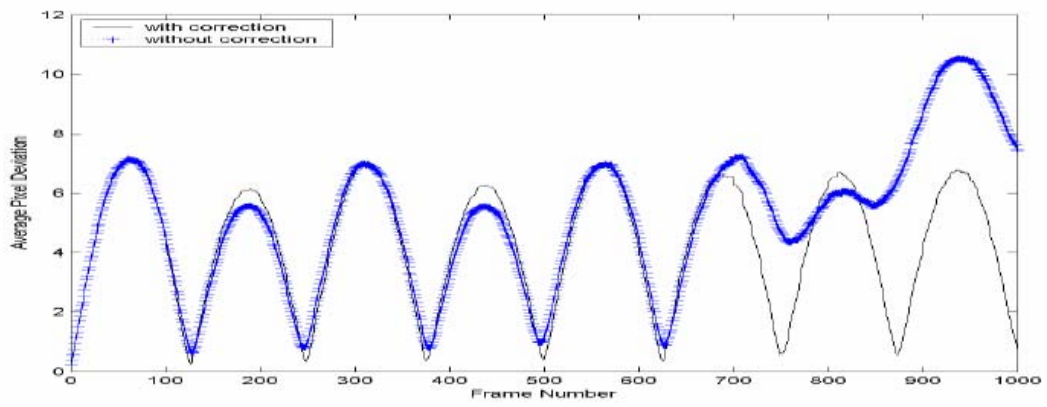
(a) Test Video 1



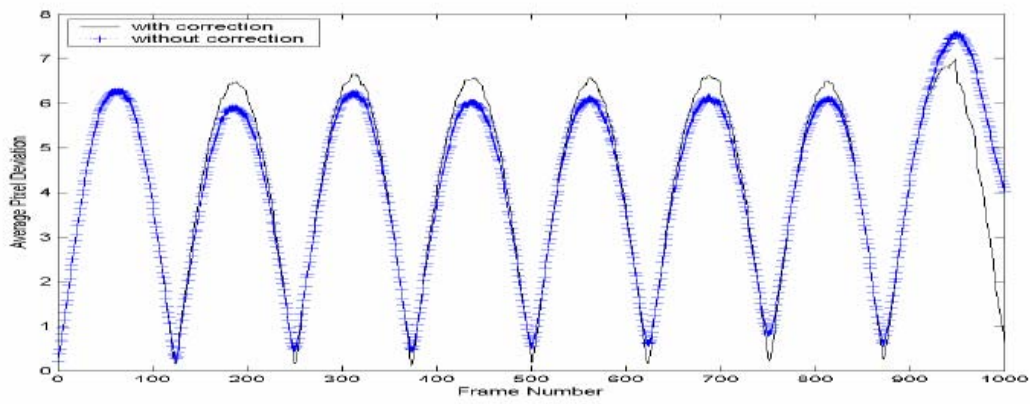
(b) Test Video 2



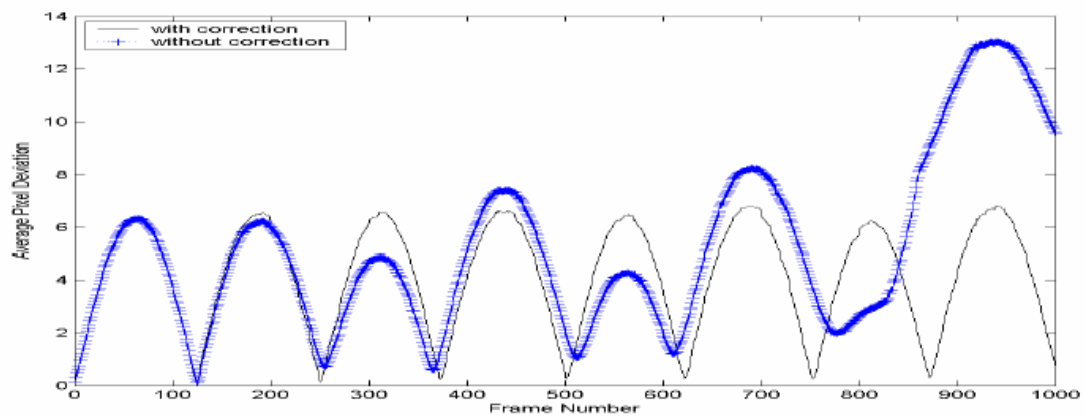
(c) Test Video 3



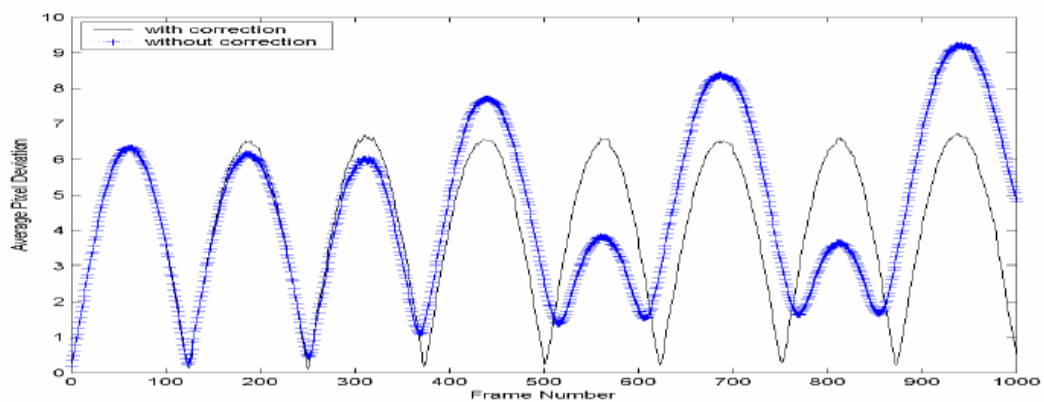
(d) Test Video 4



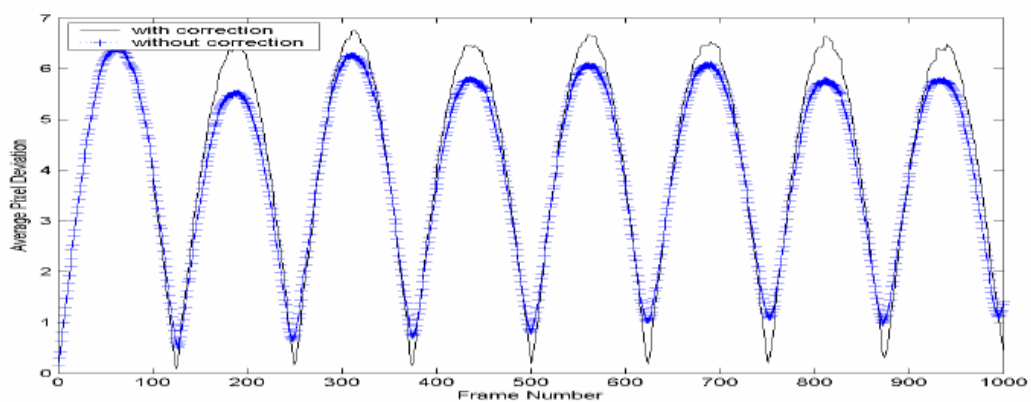
(e) Test Video 5



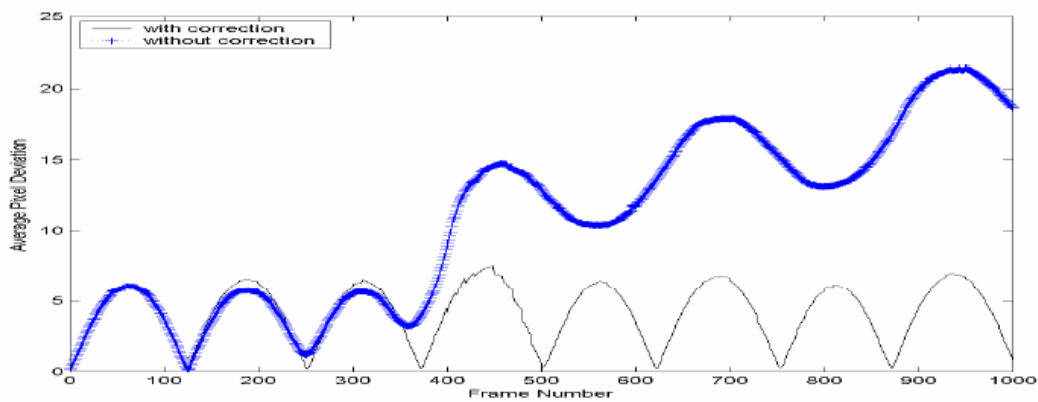
(f) Test Video 6



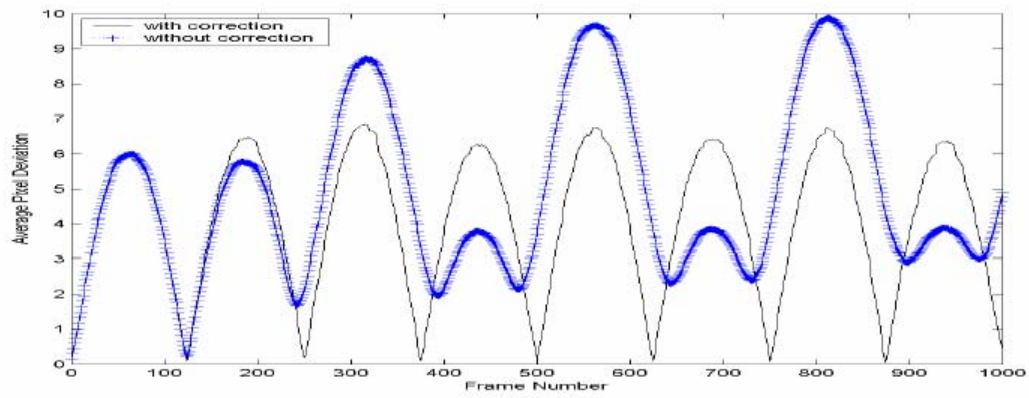
(g) Test Video 7



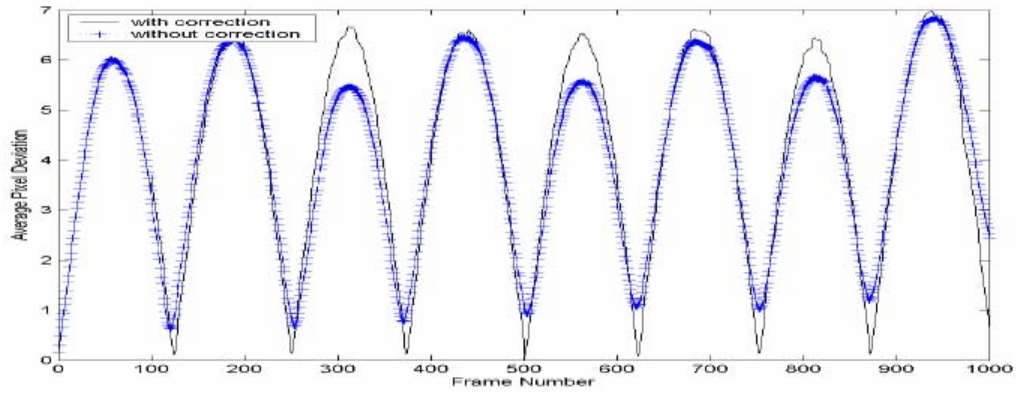
(h) Test Video 8



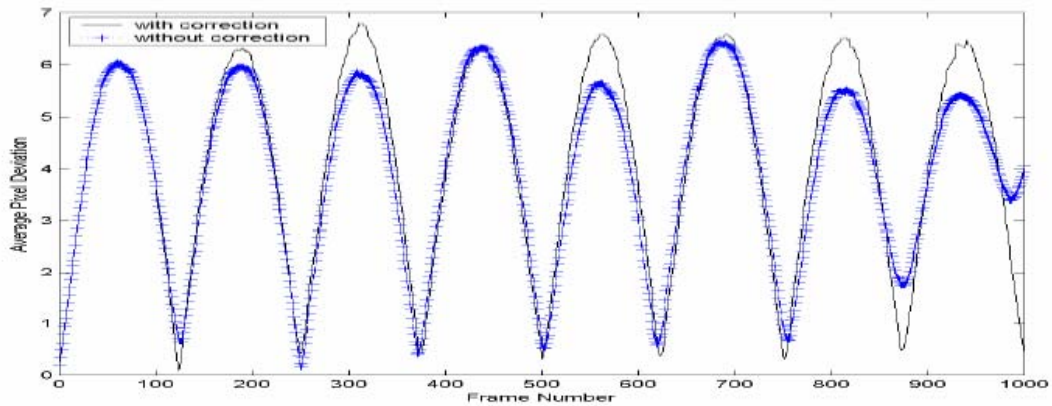
(i) Test Video 9



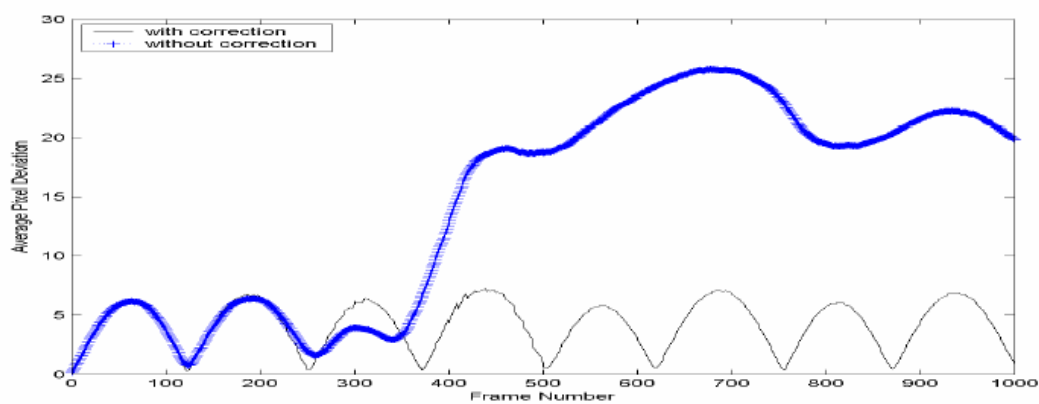
(j) Test Video 10



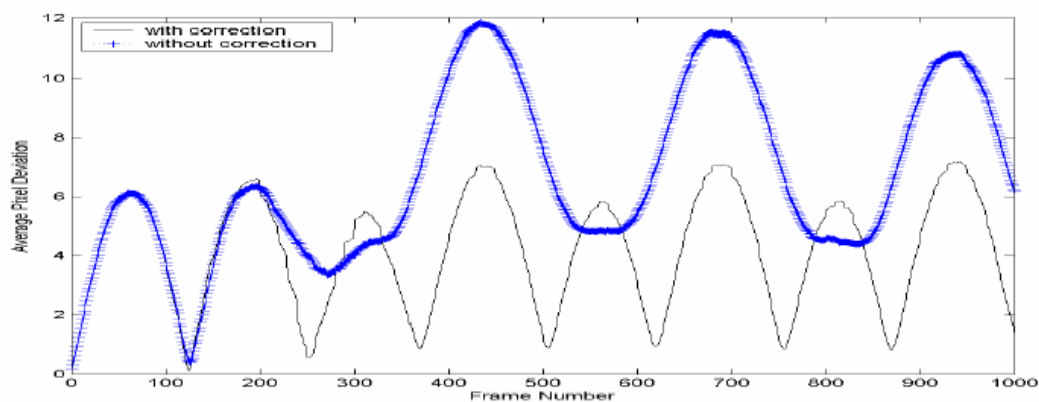
(k) Test Video 11



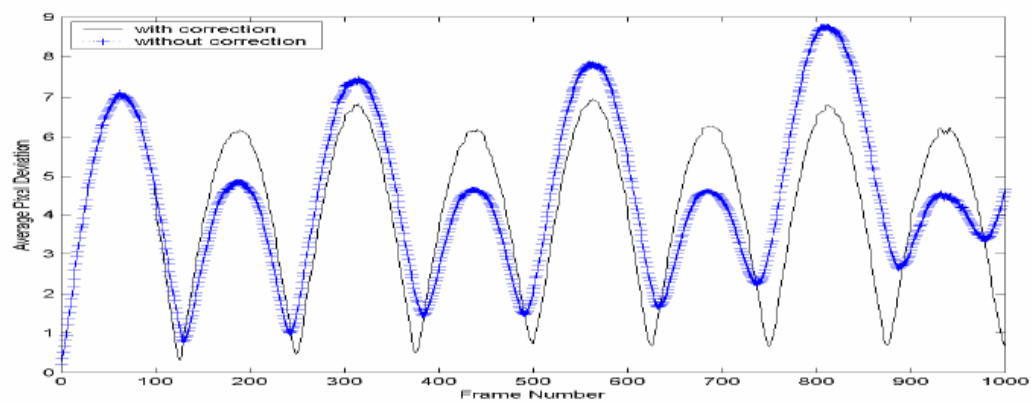
(l) Test Video 12



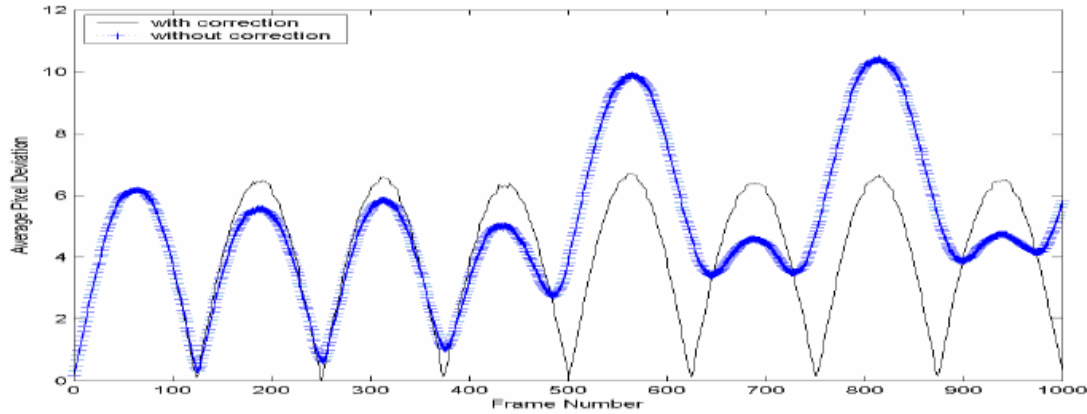
(m) Test Video 13



(n) Test Video 14



(o) Test Video 15



(p) Test Video 16

Figure 4.6 Effects of periodic correction on 16 test videos

#### 4.3.4 Evaluation of Motion Correction Methods

The general strategy of motion correction is to apply low pass filtering to the sequence of estimated motion vector in order to filter out the unwanted motions. The filtered vector for each frame is used to compensate its motion in comparison to the reference frame.

We have implemented the following motion correction methods and evaluated their performance: motion vector integration (MVI), frame position smoothing (FPS), and Kalman filtering.

##### 4.3.4.1 Motion vector integration

The estimated motion vector is integrated using a damping factor:

$$V_{\text{Integrated}}(n) = \delta * V_{\text{Integrated}}(n-1) + V_{\text{Estimated}}(n) \quad (52)$$

where  $\delta$  is the damping factor, which can range from 0.45 to 1, and  $V_{\text{Estimated}}(n)$  represents the inter-frame motion between frame  $n-1$  and frame  $n$ . Larger values of  $\delta$  are more effective in filtering out unwanted motions, but also introduce more delay in the integrated vector. This tradeoff needs to be addressed based on an analysis of the actual data, so that an appropriate value of  $\delta$  can be determined.

##### 4.3.4.2 Frame position smoothing

The estimated motion vector is low-pass filtered using a variety of approaches, such as DFT domain filtering and digital filtering. DFT domain filtering requires processing the frames off-line, therefore it is not suitable for on-line real-time video stabilization system. In contrast, digital filtering can be implemented using recursive equations:

$$y[n] = \sum_{i=0}^k a_i x[n-i] - \sum_{i=1}^k b_i y[n-i] \quad (53)$$

where  $x$  is the input data, and  $y$  represents the filtered data. For example, a Butterworth low-pass IIR digital filter of order 6 with a cut-off frequency of 0.15 (relative to the half of the sampling rate) can be implemented by the recursive equations:

$$\begin{aligned} y[n] = & 0.00007621454901 * x[n] + 0.0004572872941 * x[n-1] + \\ & 0.001143218235 * x[n-2] + 0.001524290980 * x[n-3] + \\ & 0.001143218235 * x[n-4] + 0.0004572872941 * x[n-5] + \\ & 0.00007621454901 * x[n-6] + 4.182389579 * y[n-1] - \\ & 7.491611085 * y[n-2] + 7.313595967 * y[n-3] - \\ & 4.089349932 * y[n-4] + 1.238525372 * y[n-5] - 1.584276326 * y[n-6] \end{aligned} \quad (54)$$

#### 4.3.4.3 Kalman filtering

Kalman filtering [26][27] describes a recursive solution to the discrete-time state estimation problem. It is assumed that the dynamic model of the random process at time  $t_k$  is defined as follows:

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{w}_k \quad (55)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (56)$$

where  $\mathbf{x}_k$  represents the state vector,  $\Phi_k$  denotes the state transition matrix,  $\mathbf{z}_k$  stands for the

measurement vector, and  $\mathbf{H}_k$  relates the measurement vector with the state transition matrix.  $\mathbf{w}_k$  and  $\mathbf{v}_k$  characterize the errors in the state transition process and the measurement process, respectively. They are assumed to be white noise sequences and satisfy:

$$E[\mathbf{w}_k \mathbf{w}_k^T] = \begin{cases} \mathbf{Q}_k, & i = k \\ 0, & i \neq k \end{cases} \quad (57)$$

$$E[\mathbf{v}_k \mathbf{v}_k^T] = \begin{cases} \mathbf{R}_k, & i = k \\ 0, & i \neq k \end{cases} \quad (58)$$

$$E[\mathbf{w}_k \mathbf{v}_i^T] = 0, \quad \text{for all } k \text{ and } i \quad (59)$$

At time  $t_k$ , assume that the prior estimate of the state vector is  $\hat{\mathbf{x}}_k^-$ . When the

measurement  $z_k$  becomes available, the posterior estimate  $\hat{x}_k$  is obtained by linearly blending  $\hat{x}_k^-$  and  $z_k$ :

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \quad (60)$$

In order to minimize the mean-square estimation error, represented by the trace of the posterior error covariance matrix

$$P_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)] \quad (61)$$

$K_k$  can be solved as:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (62)$$

which is also referred to as the *Kalman gain*.

The complete Kalman filtering process is listed as follows:

1) Compute the prior estimates of the state vector and the error covariance matrix based on their posterior estimates at the previous moment:

$$\hat{x}_k^- = \Phi_{k-1} \hat{x}_{k-1} \quad (63)$$

$$P_k^- = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + Q_{k-1} \quad (64)$$

2) Update the prior estimate using the measurement to obtain the posterior estimate at the current moment:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (65)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \quad (66)$$

$$P_k = (I - K_k H_k) P_k^- \quad (67)$$

where

$$P_k^- = E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)] \quad (68)$$

is the prior error covariance matrix.

#### 4.3.4.4 Test Results

We assume that the estimated motion parameters (scale factor, rotation angle, horizontal and vertical translations) are impacted by independent random effects, which are responsible for their deviations from ground truth values. Therefore, instead of applying the filtering algorithm to a motion vector, which may involve computation of matrices of high dimension, we decompose the motion vector into each individual component and apply one-dimensional filtering method to each component separately.

In this study, we assume that the intentional motion follows a synthetic linear model. For example, a camera pans with a constant speed. The intentional motion  $l_n$  is contaminated



by two types of noise, a cyclic component  $c_n$  for modeling mechanical vibration, and a random component  $r_n$  for modeling the cumulative effects of random forces. Specifically, let  $x_n$  represents the estimated (observed) value of the motion parameter:

$$x_n = l_n + c_n + r_n \quad (69)$$

where

$$l_n = \begin{cases} 0.4 * n & 1 \leq n \leq 25 \\ 20 - 0.4 * n & 26 \leq n \leq 75 \\ 0.4 * n - 40 & 76 \leq n \leq 100 \\ 0.2 * n - 20 & 101 \leq n \leq 150 \\ 40 - 0.2 * n & 151 \leq n \leq 250 \\ 0.2 * n - 60 & 251 \leq n \leq 300 \end{cases} \quad (70)$$

$$c_n = 2 \sin\left(\frac{2\pi}{10}n\right) \quad (71)$$

$$r_n \sim N(0, 1) \quad (72)$$

Figure 4.7 shows the ground truth  $l_n$  and estimated value  $x_n$  with respect to an index  $n$ , which can be interpreted as the frame number.

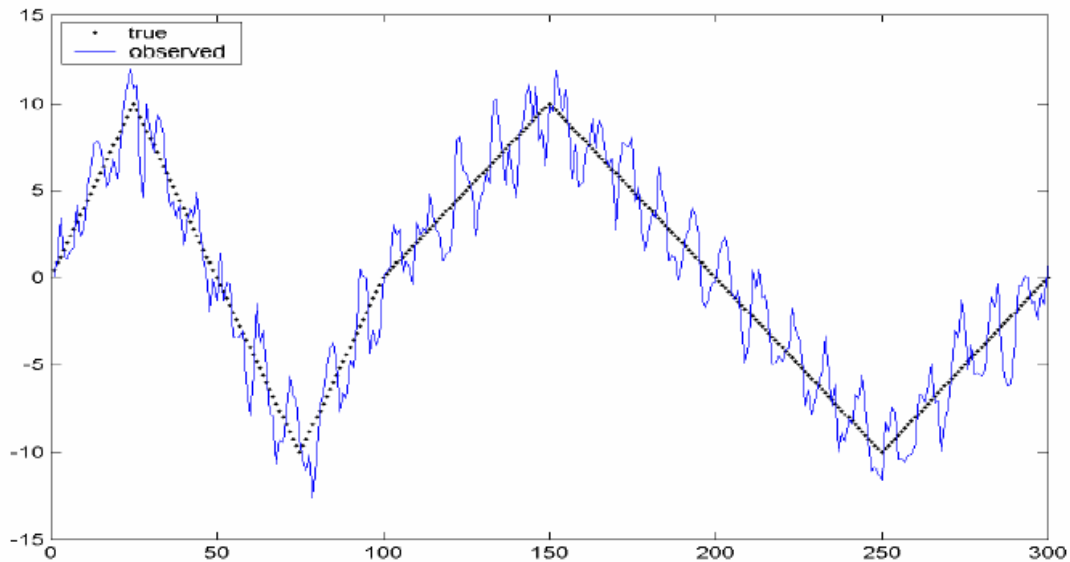
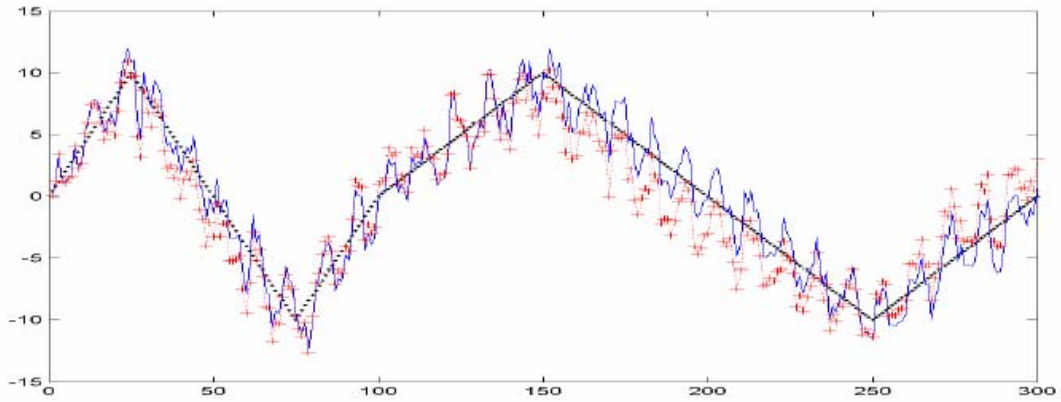


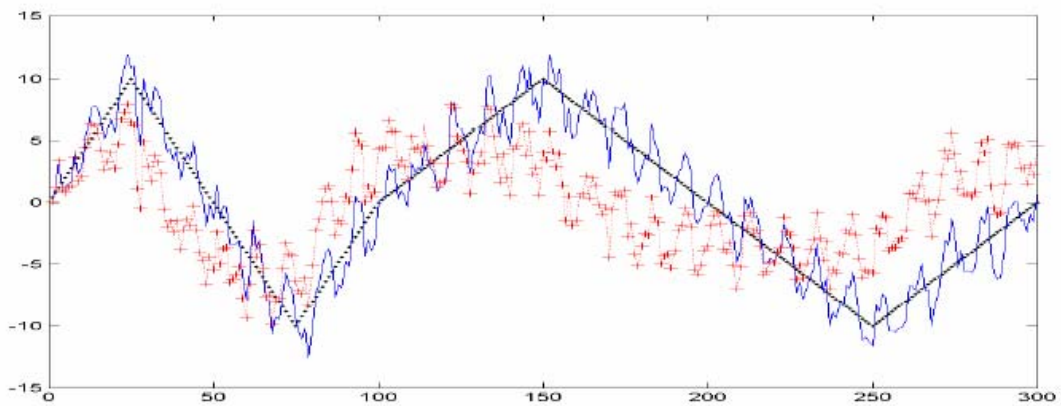
Figure 4.7 True and observed synthetic motion model

Figure 4.8 (a) - (f) shows the filtering results (curve of ``+') using motion vector integration with varying values of the parameter  $\delta$  (0.6, 0.7, 0.8, 0.9, 0.95, 0.99).

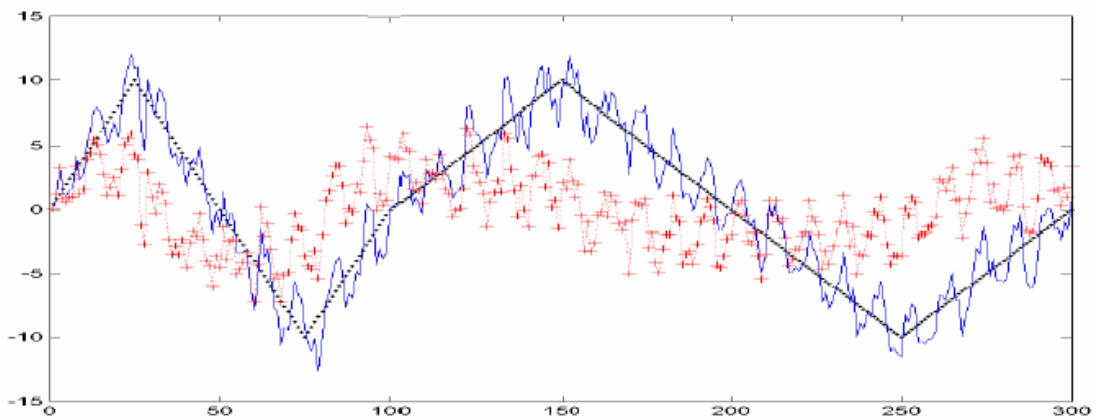
## Center for Coastline Security Technology Year Two-Final Report



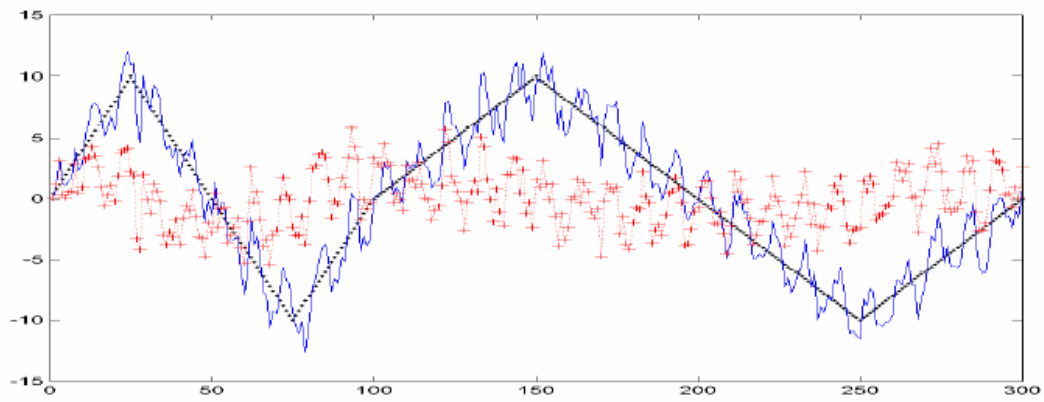
(a)



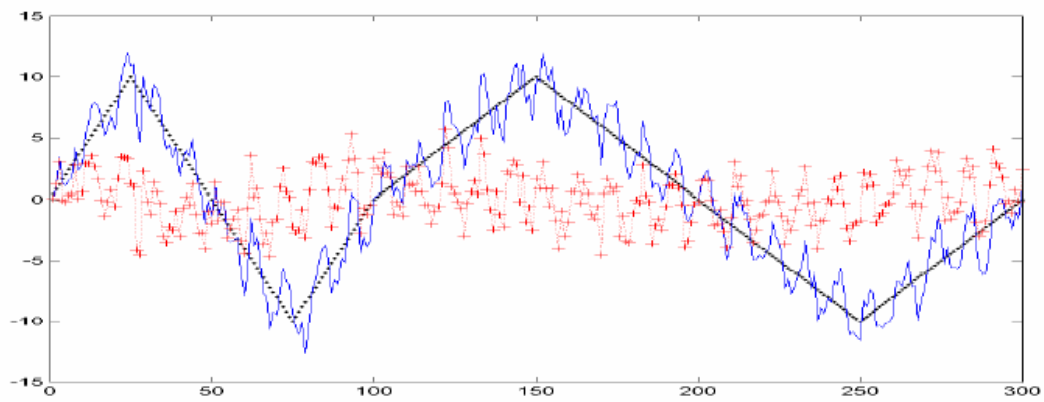
(b)



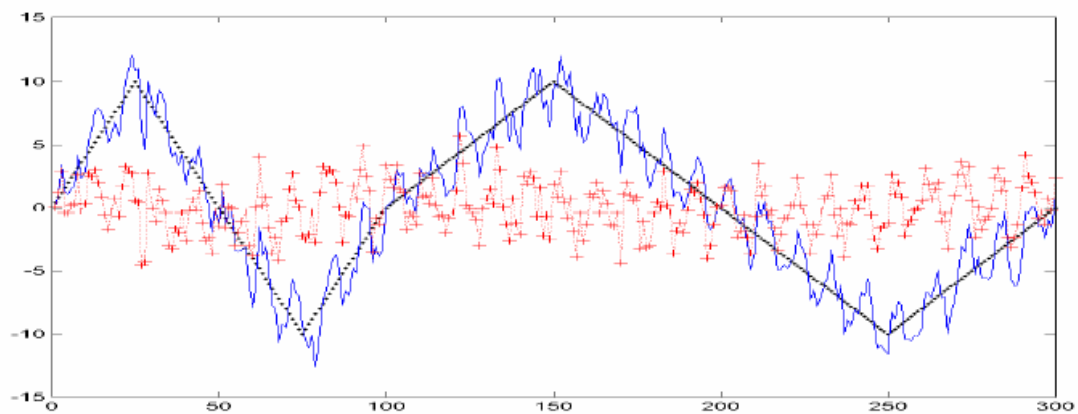
(c)



(d)



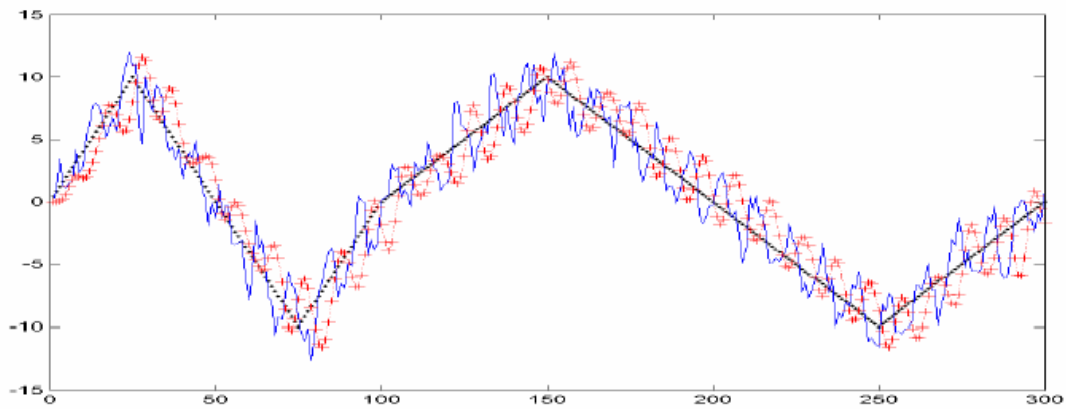
(e)



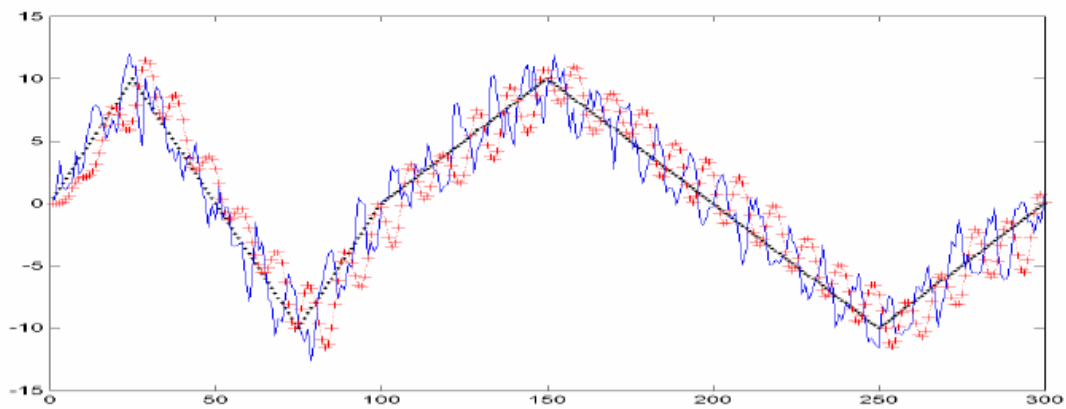
(f)

Figure 4.8 Filtering results of motion vector integration

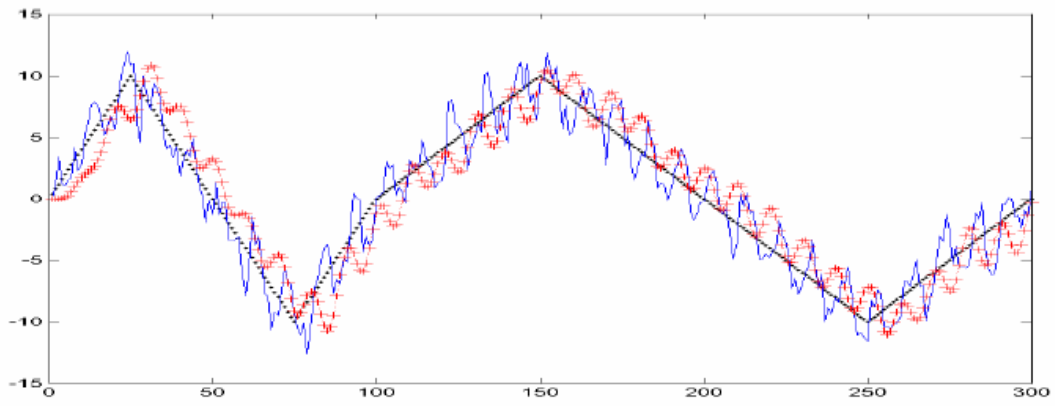
Figure 4.9 (a) - (f) shows the filtering results using frame position smoothing. The filter is a Butterworth low-pass IIR digital filter of order 6 with varying relative cut-off frequency values (0.05, 0.1, 0.15, 0.2, 0.25, 0.3).



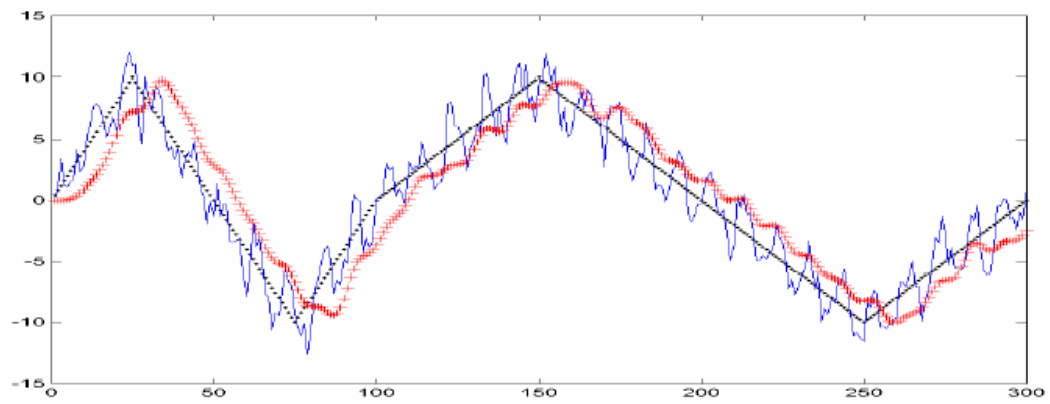
(a)



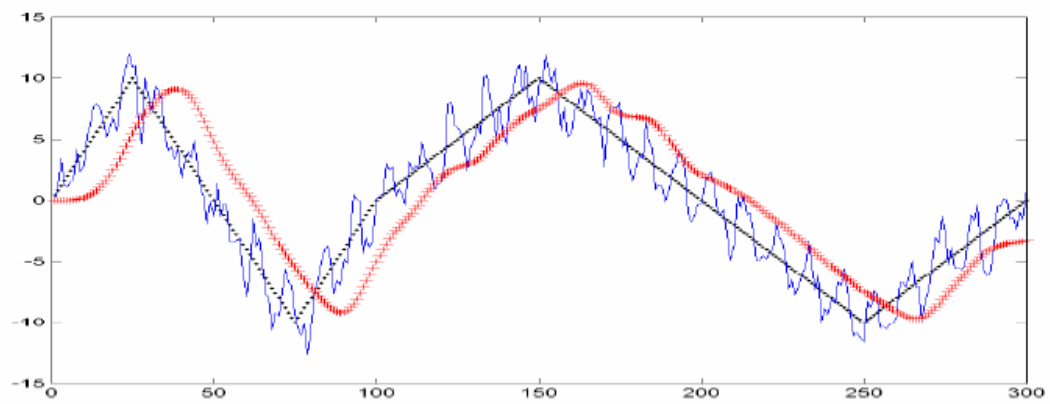
(b)



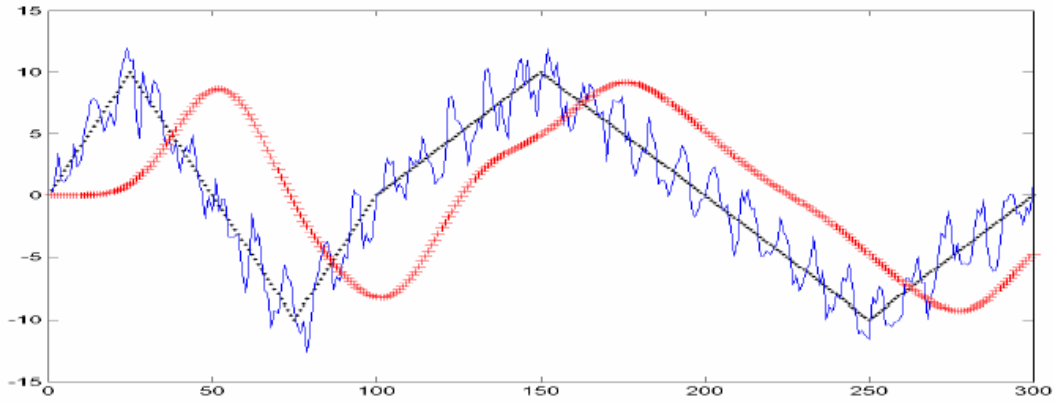
(c)



(d)



(e)

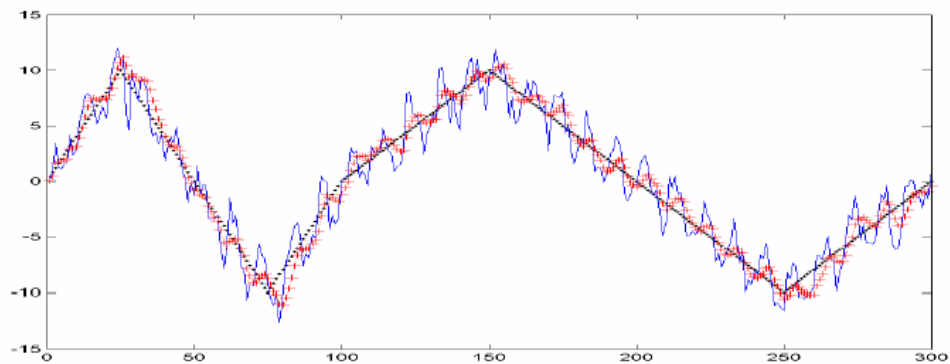


(f)

Figure 4.9 Filtering results of frame position smoothing

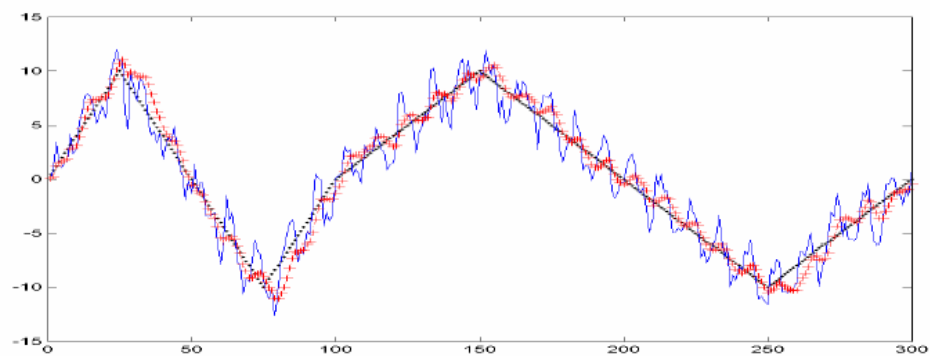
Figure 4.10 (a) - (f) shows the filtering results using Kalman filtering with varying values of the variance of the measurement error (100, 200, 500, 1000, 5000, 10000). We assume that the estimated motion parameter  $x$  has constant change rate that is subject to random noise  $r \sim N(0, \sigma)$ :

$$\begin{pmatrix} x_n \\ dx_n \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_{n-1} \\ dx_{n-1} \end{pmatrix} + \begin{pmatrix} 0 \\ r \end{pmatrix} \quad (73)$$

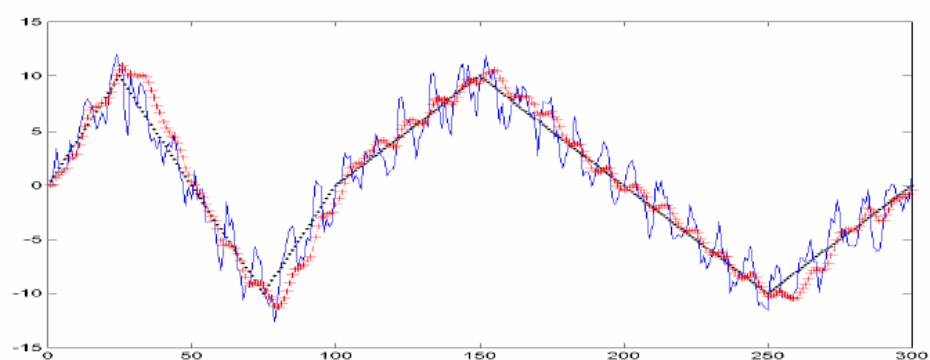


(a)

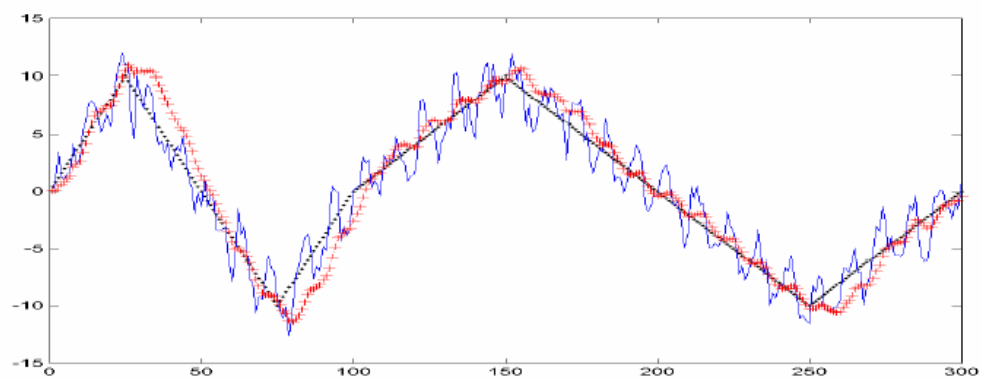
## Center for Coastline Security Technology Year Two-Final Report



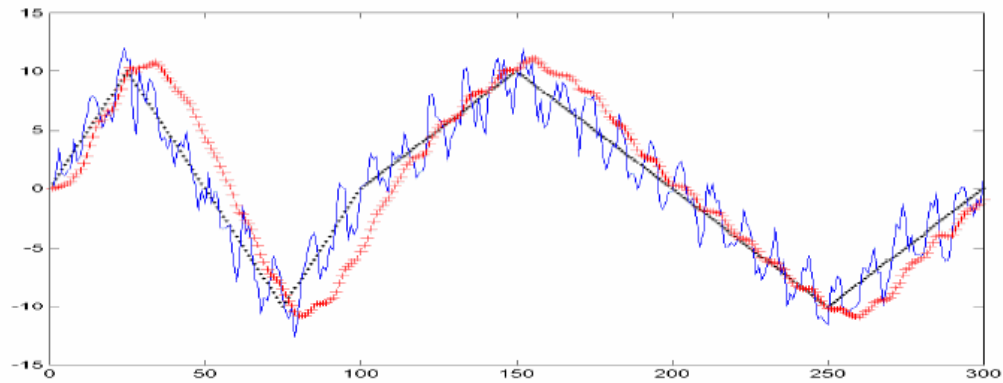
(b)



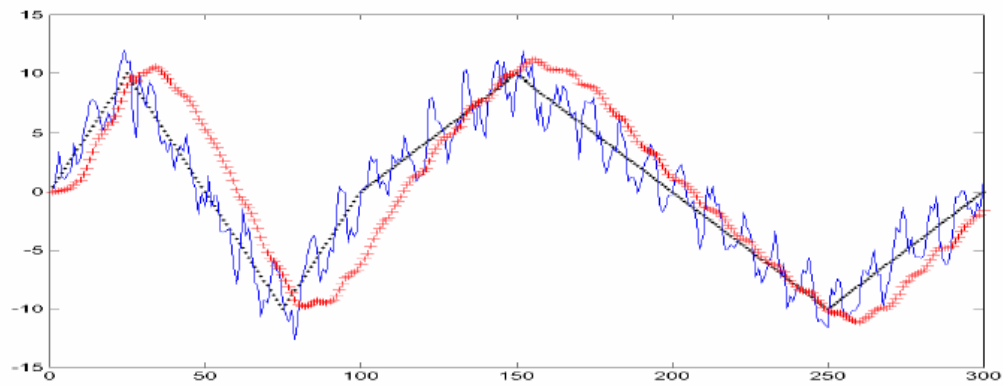
(c)



(d)



(e)



(f)

Figure 4.10 Filtering results of Kalman filtering

From the figures, both motion vector integration and frame position smoothing exhibit a tradeoff between effectiveness of smoothing and delay. At approximately equal amount of delay, frame position smoothing performs much better than motion vector integration, since it employs a more sophisticated filtering equation. The result of Kalman filtering also shows some delay at all turning points. But it is interesting to notice that the result of Kalman filtering is capable of gradually tracking the true signal and close in on it, which can be attributed to its dynamic update strategy. Therefore, Kalman filtering should be employed as the motion correction method.

#### 4.3.5 A Complete Algorithm for Video Stabilization

Based on the empirical studies performed in the above two sections, we formulate a complete algorithm for video stabilization as follows.



1) Let  $I^n$  be the image function in the current frame  $n$  that maps a location  $(x, y)$  to its intensity value. For frame  $n$ , estimate the transformation matrix  $\mathbf{T}_n^0$  between it and the reference frame:

- a) If  $n$  satisfies the condition for periodic correction (recall from subsection 3.11, this condition is satisfied every  $T$  frames), then determine frame  $m$  from Equation 46, and compute the transformation matrix for frame  $I^n$  by Equation 49.
- b) Otherwise, estimate the motion between frame  $I^{n-1}$  and frame  $I^n$  (represented by  $\mathbf{T}_n^{n-1}$ ) by KLT-based feature tracking algorithm (section 3.4), and compute the transformation matrix for frame  $n$  according to

$$\mathbf{T}_n^0 = \mathbf{T}_{n-1}^0 \mathbf{T}_n^{n-1} \quad (74)$$

2) For each of the estimated motion parameters  $x[n]$  in  $\mathbf{T}_n^0$  ( $x$  represents one of  $s, \theta, d_x$ , and

$d_y$ ), apply Kalman filtering to the sequence  $\{x[i] : 1 \leq i \leq n\}$  to obtain its filtered value. The filtered values for all the estimated motion parameters form the filtered transformation matrix

$$\hat{\mathbf{T}}_n^0 = \begin{pmatrix} \hat{s}[n] \cos(\hat{\theta}[n]) & -\hat{s}[n] \sin(\hat{\theta}[n]) & \hat{d}_x[n] \\ \hat{s}[n] \sin(\hat{\theta}[n]) & \hat{s}[n] \cos(\hat{\theta}[n]) & \hat{d}_y[n] \\ 0 & 0 & 1 \end{pmatrix} \quad (75)$$

The filtered transformation matrix represents estimated intentional motion.

3) In order to preserve intentional motion while removing unwanted motions, the

stabilized frame  $I_{\text{stabilized}}^n$  is constructed based on frame  $I^n$  by two steps. For each pixel  $(x, y)$ ,

- a) find its corresponding location in the reference frame

$$\begin{pmatrix} x^0 \\ y^0 \\ 1 \end{pmatrix} = (\mathbf{T}_n^0)^{-1} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (76)$$

- b) find its corresponding location in the current frame if there were no unwanted motion, which is equivalent to the case where there is only intentional motion

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix} = \hat{\mathbf{T}}_n^0 \begin{pmatrix} x^0 \\ y^0 \\ 1 \end{pmatrix} \quad (77)$$

Therefore, the intensity value at location  $(x, y)$  in the stabilized frame is determined by

$$I_{\text{stabilized}}^n(x, y) = I^n(\hat{x}, \hat{y}) \quad (78)$$

Usually  $(\hat{x}, \hat{y})$  is not located at integer image grid points, and a variety of interpolation methods can be employed, such as nearest neighbor interpolation, bilinear interpolation, bi-cubic interpolation, etc. Although the results of bi-cubic interpolation are slightly better than bilinear interpolation, it requires larger interpolation neighborhoods and much more computational cost. In this study, we use bilinear interpolation based on its effectiveness and low computational cost.

The intensity value at location  $(x, y)$  in the stabilized frame is determined by:

$$\begin{aligned} I_{\text{stabilized}}^n(x, y) &= I^n(\hat{x}, \hat{y}) \\ &= I^n(\lfloor \hat{x} \rfloor, \lfloor \hat{y} \rfloor) * [1 - (\hat{x} - \lfloor \hat{x} \rfloor)] * [1 - (\hat{y} - \lfloor \hat{y} \rfloor)] + \\ &\quad I^n(\lfloor \hat{x} \rfloor, \lfloor \hat{y} \rfloor + 1) * [1 - (\hat{x} - \lfloor \hat{x} \rfloor)] * [\hat{y} - \lfloor \hat{y} \rfloor] + \\ &\quad I^n(\lfloor \hat{x} \rfloor + 1, \lfloor \hat{y} \rfloor) * [\hat{x} - \lfloor \hat{x} \rfloor] * [1 - (\hat{y} - \lfloor \hat{y} \rfloor)] + \\ &\quad I^n(\lfloor \hat{x} \rfloor + 1, \lfloor \hat{y} \rfloor + 1) * [\hat{x} - \lfloor \hat{x} \rfloor] * [\hat{y} - \lfloor \hat{y} \rfloor] \end{aligned} \quad (79)$$

where  $\lfloor x \rfloor$  denotes the greatest integer less than or equal to  $x$ .

#### 4.3.6 Experimental Results

Figure 4.11 shows examples of stabilization results on the synthetic videos. In each figure, the left sub-figure shows the frame with imposed vibration, while the right sub-figure shows the stabilized frame. To demonstrate the effectiveness of the proposed video stabilization algorithm, all the examples involve imposed vibrations at nearly extreme values of the motion parameters. The corresponding APD value is around 7, which is less than 2 percent of the diagonal of the frames with a size of 300×200. The effect of stabilization can be identified by examining the orientation and size of the caption in the lower right corner in each frame.

The processing speed is about 0.09 seconds per frame on a PC with Intel 2.8GHz CPU and 512M RAM. Vast majority of the computation time is spent on the motion estimation stage, and in particular, the process of tracking features, which can be parallelized since features can be tracked independently.

## Center for Coastline Security Technology Year Two-Final Report



(a)



(b)



(c)

## Center for Coastline Security Technology Year Two-Final Report



(d)



(e)



(f)

Center for Coastline Security Technology Year Two-Final Report



(g)



(h)



(i)



(j)



## Center for Coastline Security Technology Year Two-Final Report



(k)



(l)



(m)



(n)

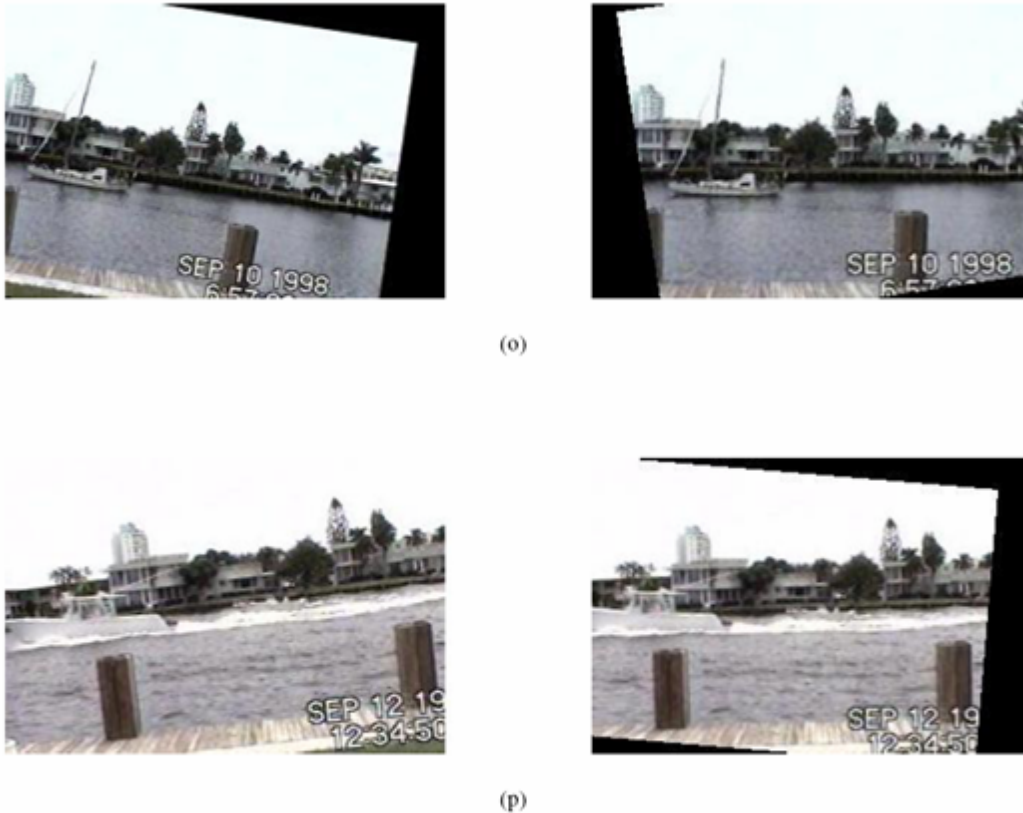


Figure 4.11 Stabilization results for test sequences

#### 4.3.7 Conclusion

The objective of video stabilization is to remove undesirable motion effects so that only intentional motion effects are retained. The primary benefit of video stabilization is improving video quality. In this report, we have performed an empirical study of software based stabilization techniques in order to develop a real-time stabilization algorithm for coastline surveillance.

Software based stabilization techniques usually operate in two stages: motion estimation and motion correction. The motion estimation stage aims to estimate the global motions between adjacent frames, which can be accumulated in order to compute the motion parameters of each frame with respect to a reference frame. The motion parameters reflect not only intentional camera motions, but also unwanted motions. The general strategy of motion correction is to apply low pass filtering to the estimated motion parameters in order to filter out the unwanted motions. The filtered parameters for each frame are used to compensate its motion with respect to the reference frame.

We have implemented four motion estimation methods and evaluated their performance: phase correlation, KLT-based feature tracking, KLT-based block matching, and integral projection. Based on computation time and accuracy, KLT-based feature tracking is the preferred method for motion estimation. We have implemented the following motion

correction methods and evaluated their performance: motion vector integration (MVI), frame position smoothing (FPS), and Kalman filtering. Kalman filtering clearly outperforms the other two methods. We have formulated a complete algorithm for video stabilization based on KLT-based feature tracking and Kalman filtering. Experimental results on synthetic videos are presented to demonstrate the effectiveness of the algorithm.

We have proposed a measure “average pixel deviation” (APD) to directly assess the accuracy of estimated motion parameters in comparison to true motion parameters. This measure is capable of overcoming the shortcomings of previous measures. In addition, we have proposed a novel periodic correction strategy to reduce error accumulation. Error accumulation is a practical issue that often arises during the computation of the transformation matrix between the current frame and the reference frame, which has not been addressed in the previous studies to the best of our knowledge.

### **References for Section 4.3**

- [1] L. Marcenaro, G. Vernazza, and C. Regazzoni, “Image stabilization algorithms for video-surveillance applications,” in Proceedings. 2001 International Conference on Image Processing, vol. 1, 2001, pp. 349–352.
- [2] A. Engelsberg and G. Schmidt, “A comparative review of digital image stabilising algorithms for mobile video communications,” IEEE Transactions on Consumer Electronics, vol. 45, no. 3, pp. 591–597, 1999.
- [3] R. Kurazume and S. Hirose, “Development of image stabilization system for remote operation of walking robots,” in Proc. Int. Conf. Robotics Automation, 2000, pp. 1856–1861.
- [4] B. Reddy and B. Chatterji, “An FFT-based technique for translation, rotation, and scale-invariant image registration,” IEEE Transactions on Image Processing, vol. 5, no. 8, pp. 1266–1271, 1996.
- [5] S. Erturk and T. Dennis, “Image sequence stabilisation based on dft filtering,” in IEE Proceedings- Vision, Image and Signal Processing, vol. 147, no. 2, 2000, pp. 95–102.
- [6] S. Auberger and C. Miro, “Digital video stabilization architecture for low cost devices,” in Proceedings of the 4<sup>th</sup> International Symposium on Image and Signal Processing and Analysis, 2005, pp. 474– 479.
- [7] H.-C. Chang, S.-H. Lai, and K.-R. Lu, “A robust and efficient video stabilization algorithm,” in IEEE International Conference on Multimedia and Expo, vol. 1, 2004, pp. 29– 32.



- [8] A. Litvin, J. Konrad, and W. Karl, "Probabilistic video stabilization using kalman filtering and mosaicking," in Proc. SPIE Image and Video Communications and Process., vol. 5022, 2003, pp. 663–674.
- [9] M. Hansen, P. Anandan, K. Dana, der van Wal, and G. B. P., "Real-time scene stabilization and mosaic construction," in Proceedings of the Second IEEE Workshop on Applications of Computer Vision, 1994, pp. 54–62.
- [10] Z. Pan and C.-W. Ngo, "Selective object stabilization for home video consumers," IEEE Transactions on Consumer Electronics, vol. 51, no. 4, pp. 1074–1084, 2005.
- [11] C. Morimoto and R. Chellappa, "Fast electronic digital image stabilization," in Proceedings of the 13th International Conference on Pattern Recognition, vol. 3, 1996, pp. 284–288.
- [12] A. Censi, A. Fusiello, and V. Roberto, "Image stabilization by features tracking," in Proceedings. International Conference on Image Analysis and Processing, 1999, pp. 665–667.
- [13] A. Crawford, H. Denman, F. Kelly, F. Pitie, and A. Kokaram, "Gradient based dominant motion estimation with integral projections for real time video stabilisation," in International Conference on Image Processing, vol. 5, 2004, pp. 3371– 3374 Vol. 5.
- [14] K. Ratakonda, "Real-time digital video stabilization for multi-media applications," in Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, vol. 4, 1998, pp. 69–72.
- [15] S. Piva, M. Zara, G. Gera, and C. Regazzoni, "Color-based video stabilization for real-time on-board object detection on high-speed trains," in IEEE Conference on Advanced Video and Signal Based Surveillance, 2003, pp. 299–304.
- [16] S. Erturk, "Image sequence stabilisation: motion vector integration (mvi) versus frame position smoothing (fps)," in Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis, 2001, pp. 266–271.
- [17] —, "Image sequence stabilisation based on kalman filtering of frame positions," Electronics Letters, vol. 37, no. 20, pp. 1217–1219, 2001.
- [18] C. Morimoto and R. Chellappa, "Evaluation of image stabilization algorithms," in Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 5, 1998, pp. 2789–2792.
- [19] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in International Joint Conference on Artificial Intelligence, 1981, pp. 674–679.

- [20] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, 1991.
- [21] J. Shi and C. Tomasi, "Good features to track," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600.
- [22] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt, "Performance of optical flow techniques," International Journal of Computer Vision, vol. 12, no. 1, pp. 43–77, 1994.
- [23] F. Hampel, P. Rousseeuw, E. Ronchetti, and W. Stahel, Robust Statistics: the Approach Based on Influence Functions. Wiley & Son, 1986.
- [24] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim, "Robust regression methods for computer vision: A review," International Journal of Computer Vision, vol. 6, no. 1, pp. 59–70, 1991.
- [25] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, vol. 24, pp. 381–395, 1981.
- [26] R. E. Kalman, "A new approach to linear filtering and prediction problems," Transactions of the ASME–Journal of Basic Engineering, vol. 82, no. Series D, pp. 35–45, 1960.
- [27] R. G. Brown and P. Y. C. Hwang, Introduction to Random Signals and Applied Kalman Filtering. John Wiley & Sons, 2001.

## **4.4 Computational Stereo: Camera Calibration**

### **4.4.1 Introduction**

Camera calibration is an important issue in computer vision since it is related to many vision problems such as stereovision, structure from motion, robot navigation and change detection. Camera calibration consists in the estimation of a model for an un-calibrated camera. The objective is to find the external parameters that describe the camera's pose (i.e. position and orientation relatively to a world co-ordinate system), and the internal parameters of the camera (principal point or image centre, focal length and distortion coefficients) that describe how the camera forms an image. Good camera calibration is important when we need to reconstruct a world model or interact with the world, e.g., robot, hand-eye coordination etc.

Camera calibration can be regarded as a least-squares parameter estimation problem, which estimates the intrinsic and extrinsic parameters that minimize the mean-squared deviation between predicted and observed image features. Least-squares parameter estimation is a fundamental technique extensively used in computer vision.

Much work for camera calibration has been done, which can be classified into two categories: photogrammetric calibration (which is performed by observing a calibration object whose geometry in 3D space is known with very good precision) and self-calibration (which does not need any calibration object, just by moving a camera in a static scene, the rigidity of the scene provides two constraints on the camera's internal parameters) [1]. Using self-calibration, if images are taken by the same camera with fixed internal parameters, correspondences between 3 images are sufficient to recover both the internal and external parameters which allow us to reconstruct 3D structure up to a similarity.

A normal computer user who performs vision tasks only from time to time will not be willing to invest money for expensive equipment. Therefore, flexibility, robustness and low cost are important considerations for camera calibration techniques [1].

Representative camera calibration techniques include Tsai's camera model [2] and [3] and Zhang's flexible technique of camera calibration [1]. A good implementation of the representative camera calibration techniques is the Caltech camera calibration toolbox [4].

### 4.4.2 Tsai's Camera Model

Tsai's camera model [2][3] is based on the pinhole model of perspective projection. Given the position of a point in 3D world coordinates the model predicts the position of the point's image in 2D pixel coordinates. Tsai's model has 11 parameters:

Five internal (or intrinsic) parameters:

- $C_x, C_y$ : coordinates of center of radial lens distortion.
- $f$ : effective focal length of the pin hole camera,
- $S_x$ , scale factor, account for any uncertainty due to imperfections in hardware timing for scanning and digitization
- $k$ , Radial lens distortion factor, a scale factor used to model radial lens distortion.

Six external (or extrinsic) parameters,

- $R_x, R_y, R_z$  - Rotation angles for the transformation between the world and camera coordinates
- $T_x, T_y, T_z$  - Translation components for the transformation between the world and camera coordinates

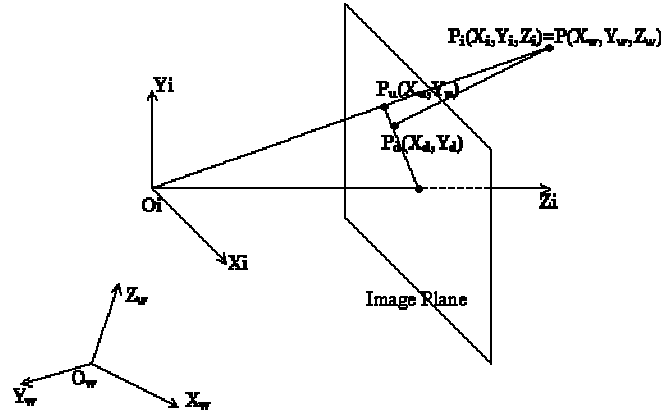


Figure 4.12 Tsai camera model with perspective projection and radial distortion

Translation:

$$t = [t_x \ t_y \ t_z]^T$$

rotation:

$$R = \begin{bmatrix} r_1 & r_2 & r_3 & 0 \\ r_4 & r_5 & r_6 & 0 \\ r_7 & r_8 & r_9 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

At the outset, there are only 3 independent rotation parameters instead of 9.  $R_x$ ,  $R_y$  and  $R_z$  are used as rotation angles for the transformation between the world and camera coordinates.

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

The transformation from a 3D point (in the image coordinate system) to the image plane is computed in the following:

1) Transform from 3D world coordinates ( $X_i, Y_i$ ) to undistorted plane ( $X_u, Y_u$ ) coordinates:

$$X_u = f \frac{X_i}{Z_i}, \quad Y_u = f \frac{Y_i}{Z_i}$$

2) Transform from undistorted ( $X_u, Y_u$ ) to distorted ( $X_d, Y_d$ ) image coordinates

$$X_u = X_d(1 + kr^2), \quad Y_u = Y_d(1 + kr^2), \quad r = \sqrt{X_d^2 + Y_d^2}$$

where  $k$  is the radial lens distortion coefficient.

3) Transform from distorted coordinates in image plane ( $X_d, Y_d$ ) to the final image coordinates ( $X_f, Y_f$ ):

$$X_f = \frac{S_x X_d}{d_x} + C_x, \quad Y_f = \frac{Y_d}{d_y} + C_y$$

where ( $d_x, d_y$ ) is the distance between adjacent camera elements in the X and Y direction.  $d_x$  and  $d_y$  are fixed parameters of the camera, which depend only on the CCD size and the image resolution, ( $X_f, Y_f$ ) are the final position in the image.

Explanations of the basic calibration algorithms and descriptions of the variables can be found in [2] and [3].

#### **4.4.3 A Flexible New Technique for Camera Calibration**

Compared with classical techniques which use expensive equipment such as 2-3 orthogonal planes, this technique is easy to use and flexible. It only requires the camera to observe a planar pattern from at least two different orientations by moving either the camera or the planar pattern (the motion does not need to be known). Radial lens distortion is modeled. The procedure of this technique consists of a closed-form solution, followed by a nonlinear refinement based on maximum likelihood criterion.

The recommended calibration procedure:

- (1) print a pattern and attach it to a planar surface;
- (2) take a few pictures of the model plane using different orientations by moving either the plane or the camera;
- (3) detect the feature points in the images;
- (4) estimate five intrinsic parameters and all the extrinsic parameters using the closed-form solution;
- (5) estimate the coefficients of the radial distortion by solving the linear least-squares;
- (6) refine all parameters by minimizing the projection error function.

The related notations and equations:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

where  $m = [u \ v]^T$  is a 2D point,  $M = [X \ Y \ Z]^T$  is a 3D point,  $s$  is an arbitrary factor,  $A$  is camera *intrinsic matrix*,

$$A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

with  $(u_0, v_0)$  the coordinates of the principal point,  $\alpha$  and  $\beta$  the scale factors in image  $u$  and  $v$  axes and  $\gamma$  the parameters describing the skewness of the two image axes.

$(R, t)$  is called *extrinsic parameters*, which includes the rotation and translation that relates the world coordinate system to the camera coordinate system.  $[\mathbf{R}, \mathbf{t}] = [r_1 \ r_2 \ r_3 \ t]$

A model point  $M$  and its image  $m$  is related by a homograph  $H$ :

$$H = [h_1 \ h_2 \ h_3] = \lambda A [r_1 \ r_2 \ t]$$

where  $\lambda$  is an arbitrary scalar.

Given one homography  $H$ , there are two basic constraints on the intrinsic parameters, which are useful to calculate the intrinsic parameters

$$\begin{aligned} h_1^T A^{-T} A^{-1} h_2 &= 0 \\ h_1^T A^{-T} A^{-1} h_1 &= h_2^T A^{-T} A^{-1} h_2 \end{aligned}$$

Once  $A$  is known, the **extrinsic parameters** for each image can be computed by

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{A}^{-1} \mathbf{h}_1 / \|\mathbf{A}^{-1} \mathbf{h}_1\| \\ \mathbf{r}_2 &= \mathbf{A}^{-1} \mathbf{h}_2 / \|\mathbf{A}^{-1} \mathbf{h}_2\| \\ \mathbf{r}_3 &= \mathbf{r}_1 \times \mathbf{r}_2 \\ \mathbf{t} &= \lambda \mathbf{A}^{-1} \mathbf{h}_3 = \mathbf{A}^{-1} \mathbf{h}_3 / \|\mathbf{A}^{-1} \mathbf{h}_1\| \end{aligned}$$

Dealing with radial distortion:

$$\begin{bmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \hat{u}-u \\ \hat{v}-v \end{bmatrix}$$

or in the matrix form as  $Dk = d$ , where  $k = [k_1, k_2]^T$ ,  $k_1$  and  $k_2$  are the coefficients of the radial distortion. The linear least-squares solution is given by

$$k = (D^T D)^{-1} D^T d$$

once  $k_1$  and  $k_2$  are estimated, we can refine the estimation of the other parameters by solving

$$\sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(A, R_i, t_i, M_j)\|^2$$

with  $\hat{m}(A, R_i, t_i, M_j)$  (the projection of point  $M_j$  in image  $i$ ) replaced by equations from the above radial distortion matrix calculation. These two procedures can be alternated until convergence.

With an example on their camera calibration website

(<http://research.microsoft.com/users/zhang/Calib/#Calibration>), Zhang [1] has the calibration result as: the pixel is *square* (aspect ratio = 1); the focal length = 832.5 pixels; the image center is at (303.959, 206.585); there is a significant radial distortion:  $k_1 = -0.228601$ ,  $k_2 = 0.190353$ . The format of the calibration file is:  $\alpha$ ,  $c$ ,  $\beta$ ,  $u_0$ ,  $v_0$ ,  $k_1$ ,  $k_2$ , then the rotation matrix and translation vector for the first image, the rotation matrix and translation vector for the second image, etc.

$\alpha$	$c$	$\beta$	$u_0$	$v_0$	$k_1$	$k_2$
832.5	0.204494	832.53	303.959	206.585	-0.228601	0.190353

[0.992759 -0.026319 0.117201]	(translation vector)
-------------------------------	----------------------

[0.0139247 0.994339 0.105341	(rotation matrix)
-0.11931 -0.102947 0.987505	
-3.84019 3.65164 12.791]	

### 4.4.4 Camera Calibration Using the Caltech Toolbox

#### 4.4.4.1 System Requirements

This toolbox works on Matlab 5.x and Matlab 6.x (up to Matlab 7.x) on Windows, Unix and Linux systems (platforms it has been fully tested) and does not require any specific Matlab toolbox. The toolbox should also work on any other platform supporting Matlab 5.x and 6.x.

#### 4.4.4.2 Getting Started

- Go to the download page:  
[http://www.vision.caltech.edu/bouguetj/calib\\_doc/download/TOOLBOX\\_calib.zip](http://www.vision.caltech.edu/bouguetj/calib_doc/download/TOOLBOX_calib.zip), and retrieve the latest version of the complete camera calibration toolbox for Matlab.
- Store the individual matlab files (.m files) into a unique folder *TOOLBOX\_calib* (default folder name).
- Run Matlab and add the location of the folder *TOOLBOX\_calib* to the main matlab path.
- Run the main matlab calibration function *calib\_gui* (or *calib*). A mode selection window appears on the screen:

Click the standard mode, the main calibration toolbox window appears on the screen



Figure 4.13 Main menu of Camera Calibration Toolbox – Standard Version

Now we are ready to use the toolbox for calibration:

#### 4.4.4.3 Calibration Example

Based on a total of 20 (and 25) images of a planar checkerboard, this example demonstrates how to use all the features of the toolbox: loading calibration images, extracting image corners, running the main calibration engine, displaying the results, controlling accuracies, adding and suppressing images, undistorting images, exporting calibration data to different formats.

- 1) Store the images into a separate folder named `calib_example`
- 2) Click the *Image names* button in the window, enter the basename of the calibration images and the image format (such as `tif`, `jpg`)
- 3) Click on the *Extract grid corners* in the window.
- 4) Main Calibration step: After corner extraction, click on the button *Calibration* of the window to run the main camera calibration procedure.
  - Calibration is done in two steps: first initialization, and then nonlinear optimization.
  - The initialization step computes a closed-form solution for the calibration parameters based on not including any lens distortion (program name: `init_calib_param.m`).
  - The non-linear optimization step minimizes the total reprojection error (in the least squares sense) over all the calibration parameters (9 DOF for intrinsic: focal, principal point, distortion coefficients, and 6\*20 DOF extrinsic => 129 parameters).
  - The optimization is done by iterative gradient descent with an explicit (closed-form) computation of the *Jacobian* matrix (program name: `go_calib_optim.m`).

Example:

Calibration parameters after initialization:

Focal Length:  $fc = [671.13759 \quad 680.77186]$   
 Principal point:  $cc = [319.50000 \quad 239.50000]$   
 Skew:  $\alpha_c = [0.00000] \implies \text{angle of pixel} = 90.00000 \text{ degrees}$   
 Distortion:  $kc = [0.00000 \quad 0.00000 \quad 0.00000 \quad 0.00000 \quad 0.00000]$

Calibration results after optimization (with uncertainties)

Focal Length:  $fc = [661.67001 \quad 662.82858] \pm [1.17913 \quad 1.26567]$   
 Principal point:  $cc = [319.50000 \quad 239.50000] \pm [2.38443 \quad 2.17481]$   
 Skew:  $\alpha_c = [0.000] \pm [0.000] \implies \text{angle of pixel} = 90.000 \pm 0.0 \text{ degrees}$   
 Distortion:  $kc = [-0.26425 \quad 0.22645 \quad 0.00020 \quad 0.00023 \quad 0.00000]$   
 $\pm [0.00934 \quad 0.03826 \quad 0.00052 \quad 0.00053 \quad 0.00000]$   
 Pixel error:  $err = [0.45330 \quad 0.38916]$



Note: the numerical errors are approximately three times the standard deviations

Notice that the skew coefficient  $\alpha_c$  and the 6th order radial distortion coefficient (the last entry of  $kc$ ) have not been estimated (this is the default mode). Therefore, the angle between the x and y pixel axes is 90 degrees. This is a practically good assumption.

Only 11 gradient descent iterations are required in order to reach the minimum. This means only 11 evaluations of the reprojection function and Jacobian computation and inversion.

Ignore the recommendation of the system to reduce the distortion model. The reprojection error is still too large to make a judgment on the complexity of the model. This is mainly because some of the grid corners were not very precisely extracted for a number of images. Thus we may need and actually we can reproject on the images.

5) Click on *Reproject on images* in the window to show the reprojections of the grids onto the original images. Then use *Show Extrinsic* in the window, and *Recom.corners* button of the tool window.

Calibration results after optimization (with uncertainties):

Focal Length:  $fc = [657.39535 \quad 657.76309] \pm [0.34691 \quad 0.37111]$

Principal point:  $cc = [302.98368 \quad 242.61630] \pm [0.70546 \quad 0.64553]$

Skew:  $\alpha_c = [0.000] \pm [0.000] \implies \text{angle of pixel} = 90.000 \pm 0.0 \text{ degrees}$

Distortion:  $kc = [-0.25584 \quad 0.12758 \quad -0.00021 \quad 0.00003 \quad 0.00000] \pm [0.00271 \quad 0.01076 \quad 0.00015 \quad 0.00014 \quad 0.00000]$

Pixel error:  $err = [0.12668 \quad 0.12604]$

Note: the numerical errors are approximately three times the standard deviations

(The re-projecting steps can be repeated several times.) This time only six iterations were necessary for convergence, and no initialization step was performed (the optimization started from the previous calibration result). The two values *0.12668* and *0.12604* are the standard deviation of the reprojection error (in pixel) in both x and y directions respectively. The numerical uncertainties values are approximately three times the standard deviations. After optimization, we can click on *Save* to save the calibration results (intrinsic and extrinsic) in the matlab file *Calib\_Results.mat*

#### 6) Compute extrinsic parameters

Use an image that was not used in the main calibration procedure. The goal is to compute the extrinsic parameters attached to this image given the intrinsic camera parameters previously computed. Click on *Comp. Extrinsic* in the window, and enter the image name without extension and the image type (*tif*), and extract the grid corners (following the same procedure as previously presented - note: the first clicked point is the origin of the pattern reference frame).

Extrinsic parameters:

Translation vector:  $Tc\_ext = [-94.617156 \quad -184.010867 \quad 766.209711]$

Rotation vector:  $omc\_ext = [-1.451113 \quad -1.827059 \quad -0.179105]$

Rotation matrix:  $Rc\_ext = \begin{bmatrix} -0.043583 & 0.875946 & -0.480436 \\ 0.765974 & 0.338032 & 0.546825 \\ 0.641392 & -0.344170 & -0.685684 \end{bmatrix}$

Pixel error:  $err = [0.10156 \quad 0.08703]$

#### 7) Undistort images

Generate the undistorted version of one or multiple images given pre-computed intrinsic camera parameters. Click on *Undistort image* in the window.

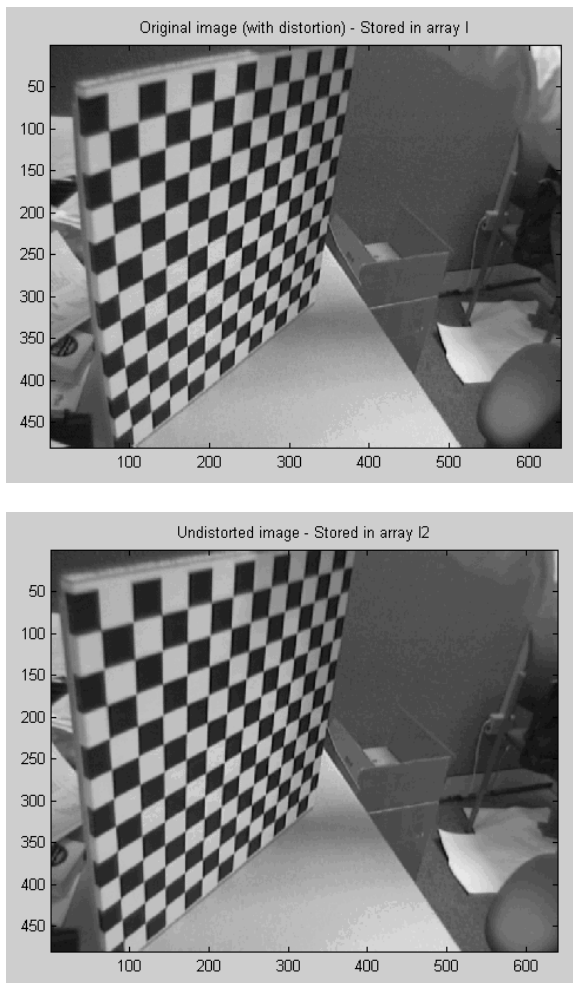


Figure 4.14 Output of Undistort Image functionality: original image (top) and undistorted image (bottom)

#### 4.4.4.4 About the Camera parameters

##### 4.4.4.4.1 Intrinsic parameters (camera model)

The internal camera model is implemented using the paper [5]. The list of internal parameters:

- *Focal length*: The focal length in pixels is stored in the 2x1 vector  $fc$ .
- *Principal point*: The principal point coordinates are stored in the 2x1 vector  $cc$ .
- *Skew coefficient*: The skew coefficient defining the angle between the x and y pixel axes is stored in the scalar  $alpha\_c$ .
- *Distortions*: The image distortion coefficients (radial and tangential distortions) are stored in the 5x1 vector  $kc$ .

Definition of the intrinsic parameters:

Let  $P$  be a point in space of coordinate vector  $XX_c = [X_c; Y_c; Z_c]$  in the camera reference frame. Let us project now that point on the image plane according to the intrinsic parameters ( $fc$ ,  $cc$ ,  $alpha\_c$ ,  $kc$ ). Let  $x_n$  be the normalized (pinhole) image projection:

$$x_n = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Let  $r^2 = x^2 + y^2$ , after including lens distortion, the new normalized point coordinate  $x_d$  is defined:

$$x_d = \begin{bmatrix} x_d(1) \\ x_d(2) \end{bmatrix} = \left( 1 + kc(1)r^2 + kc(2)r^4 + kc(5)r^6 \right) x_n + dx$$

where  $dx$  is the tangential distortion vector:

$$dx = \begin{bmatrix} 2 kc(3) x y + kc(4) (r^2 + 2x^2) \\ kc(3) (r^2 + 2y^2) + 2 kc(4) x y \end{bmatrix}$$

The 5-vector  $kc$  contains both radial and tangential distortion coefficients (the coefficient of 6<sup>th</sup> order radial distortion term is the fifth entry of the vector  $kc$ ).

This distortion model was first introduced by Brown in 1966 and was called "*Plumb Bob*" model (radial polynomial + "*thin prism*"). The tangential distortion is due to "decentering", or imperfect centering of the lens components and other manufacturing defects in a compound lens. For more details, refer to Brown's original publications [6].

The final pixel coordinates  $x_{pixel} = [x_p; y_p]$  of the projection of  $P$  on the image plane is:

$$\begin{cases} x_p = fc(1) (x_d(1) + alpha\_c * x_d(2)) + cc(1) \\ y_p = fc(2) x_d(2) + cc(2) \end{cases}$$

The pixel coordinate vector  $x_{pixel}$  and the normalized (distorted) coordinate vector  $x_d$  are related to each other through the linear equation:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = KK \begin{bmatrix} x_d(1) \\ x_d(2) \\ 1 \end{bmatrix}$$

where  $KK$  is known as the camera matrix defined as:

$$KK = \begin{bmatrix} fc(1) & \alpha_c * fc(1) & cc(1) \\ 0 & fc(2) & cc(2) \\ 0 & 0 & 1 \end{bmatrix}$$

$fc(1)$  and  $fc(2)$  are the focal distance (a unique value in mm) expressed in units of horizontal and vertical pixels. The aspect  $ratio = fc(2)/fc(1)$  is different from 1 if the pixel in the CCD array are not square. Therefore, the camera model naturally handles non-square pixels. In addition, the coefficient  $\alpha_c$  encodes the angle between the x and y sensor axes. Pixels are allowed to be non-rectangular. Some authors refer to that type of model as "affine distortion" model.

In addition to computing estimates for the intrinsic parameters  $fc$ ,  $cc$ ,  $kc$  and  $\alpha_c$ , the toolbox also returns estimates of the uncertainties on those parameters. The matlab variables containing those uncertainties are  $fc\_error$ ,  $cc\_error$ ,  $kc\_error$ ,  $\alpha_c\_error$ . Those vectors are approximately three times the standard deviations of the errors of estimation.

**Convention:** Pixel coordinates are defined as:

$[0;0]$  is the center of the upper left pixel of the image. As a result,

$[n_x-1;0]$  is center of the upper right corner pixel,

$[0;n_y-1]$  is the center of the lower left corner pixel and

$[n_x-1;n_y-1]$  is the center of the lower right corner pixel

where  $n_x$  and  $n_y$  are the width and height of the image (for the images of the first example,  $n_x=640$  and  $n_y=480$ ).

**Reduced camera models:** Currently manufactured cameras are customary to assume rectangular pixels, and thus assume zero skew ( $\alpha_c=0$ ). Furthermore, the very generic (6th order radial + tangential) distortion model is often not considered completely. For standard field of views (non wide-angle cameras), it is often not necessary (and not recommended) to push the radial component of distortion model beyond the 4th order (i.e. keeping  $kc(5)=0$ ). In addition, the tangential component of distortion can often be discarded (justified by the fact that most lenses currently manufactured do not have imperfection in centering). In the second order symmetric radial distortion model (a common distortion model, especially when only a few images are used for calibration), only the first component of the vector  $kc$  is estimated, while the other four are set to zero. Other possible model reductions: when only a few images are used for calibration (e.g. 2 or 3 images) the principal point  $cc$  is often very difficult to estimate reliably, thus it is one of the most difficult parts of the native perspective

projection model to estimate besides lens distortions. Thus, it is better to set the principal point at the center of the image ( $cc = [(nx-1)/2; (ny-1)/2]$ ) and not estimate it anymore.

#### 4.4.4.4.2 Extrinsic parameters

The list of external parameters:

- *Rotations*: A set of  $n\_ima$  3x3 rotation matrices  $Rc\_1, Rc\_2, \dots, Rc\_20$  (assuming  $n\_ima=20$ ).
- *Translations*: A set of  $n\_ima$  3x1 vectors  $Tc\_1, Tc\_2, \dots, Tc\_20$  (assuming  $n\_ima=20$ ).

Definition of the extrinsic parameters:

Consider the calibration grid # $i$  (attached to the  $i$ -th calibration image), and concentrate on the camera reference frame attached to that grid.

Without loss of generality, take  $i = 1$ . Figure 4.15 shows the reference frame ( $O, X, Y, Z$ ) attached to that calibration grid.

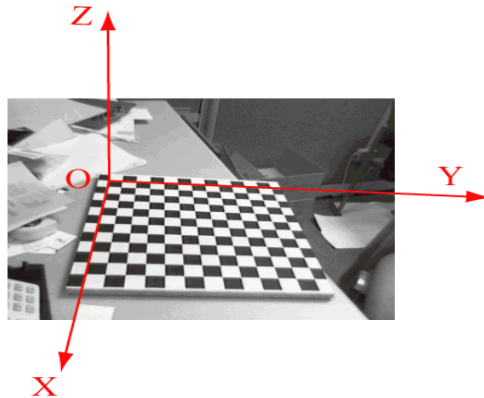


Figure 4.15 The reference frame attached to a calibration grid

Let  $P$  be a point space of coordinate vector  $XX = [X; Y; Z]$  in the grid reference frame (reference frame shown on the previous figure).

Let  $XX_c = [X_c; Y_c; Z_c]$  be the coordinate vector of  $P$  in the camera reference frame.

Then  $XX$  and  $XX_c$  are related to each other through the following rigid motion equation:

$$XX_c = Rc\_1 * XX + Tc\_1$$

The translation vector  $Tc\_1$  is the coordinate vector of the origin of the grid pattern ( $O$ ) in the camera reference frame, and the 3<sup>rd</sup> column of the matrix  $Rc\_1$  is the surface normal vector of the plane containing the planar grid in the camera reference frame.

The same relation holds for the remaining extrinsic parameters ( $Rc\_2, Tc\_2$ ), ( $Rc\_3, Tc\_3$ ), ..., ( $Rc\_20, Tc\_20$ ). Once the coordinates of a point is expressed in the camera reference frame, it may be projected on the image plane using the intrinsic

camera parameters. The vectors  $omc\_1, omc\_1, \dots, omc\_20$  are the rotation vectors associated to the rotation matrices  $Rc\_1, Rc\_1, \dots, Rc\_20$ . The two are related through the rodrigues formula, e.g.,  $Rc\_1 = rodrigues(omc\_1)$ .

Similarly to the intrinsic parameters, the uncertainties attached to the estimates of the extrinsic parameters  $omc\_i, Tc\_i$  ( $i=1, \dots, n\_ima$ ) are also computed. Those uncertainties are stored in the vectors  $omc\_error\_1, \dots, omc\_error\_20, Tc\_error\_1, \dots, Tc\_error\_20$  (assuming  $n\_ima = 20$ ) and represent approximately three times the standard deviations of the errors of estimation.

#### 4.4.5 Camera Calibration Using 3D Calibration Object

Camera calibration using 3D calibration object is performed by observing a calibration object whose geometry in 3D space is known with very good precision. A calibration object for 3D calibration usually consists of two or three planes orthogonal to each other, e.g. calibration cube, and it can also be done with a plane undergoing a precisely known translation.

The target for 3D calibration is 2 planes at right angle with checkerboard patterns (Tsai grid). Given the positions of pattern corners only with respect to a coordinate system of the target, we position the camera in front of target and find images of corners, and then obtain equations that describe point coordinates and contain intrinsic and extrinsic parameters of camera.

The main steps of 3D camera calibration:

- 1) Detect points of interest (e.g., corners of the checker pattern) in the 2D image and obtain their corresponding 3D measurement;
- 2) Find the best projection matrix  $M$  using linear least squares;
- 3) Calculate intrinsic and extrinsic parameters;
- 4) Refine the parameters through nonlinear optimization.

#### References for Section 4.4

[1] Z. Zhang, A Flexible New Technique for Camera Calibration, Technical Report, Microsoft Research, MSR-TR-98-71, 1998.

[2] R. Y. Tsai, An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, 1986, pages 364-374.

[3] Roger Y. Tsai, A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August 1987, pages 323-344.

[4] <http://www.vision.caltech.edu>

[5] J. Heikkilä and O. Silvén, A Four-step Camera Calibration Procedure with Implicit Image Correction, CVPR, 1997.

[6] D.C. Brown, Close-Range Camera Calibration, Photogrammetric Engineering, 37(8), pp. 855-866, 1971.

[7] O.D. Faugeras, Three-dimensional Computer Vision: a Geometric Viewpoint. MIT Press, Boston, 1993.

## 4.5 Stereo Correspondence

### 4.5.1 Introduction

Stereo correspondence is an active research area in computer vision. The main task of stereo correspondence is to find the disparity map between a pair of images taken from two different orientations on the same scene. Stereo matching remains a difficult vision problem, especially for textureless regions, disparity discontinuity and occlusions [1].

Local stereo matching methods (window-based) capture disparities only using intensity values within a finite neighboring window. Global stereo correspondence methods such as graph cut [2] and belief propagation [1] are used to optimize the disparity map through various minimization techniques of energy that considers matching cost, disparity discontinuities and occlusion.

For local stereo matching, small-window based matching can more accurately capture disparity in densely-textured regions than big-window based matching, but produce noisy disparities in textureless regions; while big-window matching produces smooth disparities in textureless regions, but has difficulty getting accurate disparities for densely-textured regions. Some algorithms have been proposed to capture disparity values for densely-textured regions, such as variable windows [4], and rod-shaped shiftable windows [5].

In an attempt to accurately match stereo for both densely-textured and textureless regions, we propose a progressive edge-based stereo matching method, in which the edges are extracted from the disparity map of a big-window based matching. For the regions around the edges, we use the disparity values from a small-window and arbitrarily-shaped window based matching; for the textureless regions away from the edges, we use strips of disparity values from a big-window and arbitrarily-shaped window based matching, and enforce disparity continuity between the strips. This process of unifying disparity maps from matches of differently-sized windows can be repeated progressively from a big window matching towards a smallest window matching.

The arbitrarily-shaped window based stereo matching, which uses a 5-pixel window that has arbitrary shapes and orientations, makes stereo correspondence for the regions where a regular small window or big window matching can not find matches, especially for those with fine details.

Instead of using energy minimization based optimization, we propose a local optimization method called *Progressive Outlier Remover (POR)* for the disparity map.

When a disparity value is surrounded by different disparities, it will be replaced by its neighbors' average disparity when certain conditions are met. We progressively vary the distance values between the active point and its neighbors and use threshold values to avoid over-pruning.

We work on the *Middlebury* stereo data and evaluate the performance of our algorithm in terms of the accuracy for all regions, non-occluded regions and disparity discontinuity regions of the resulting disparity maps against the ground-truth according to the *Middlebury* test bed [6].

## 4.5.2 Stereo Correspondence Algorithms

### 4.5.2.1 Local Stereo Matching

Local stereo correspondence methods use the intensity values of pixels to make stereo matching. Basically, a local stereo matching method seeks to estimate disparity at a pixel in one image (reference image) by comparing a small region (usually a square window) with a series of same-sized-and-shaped regions in the other image (matching image) along the same scanline (the horizontal axis). The correspondence between a pixel  $(x, y)$  in reference image  $R$  and a pixel  $(x', y')$  in the matching image  $M$  is given by

$$x' = x + \text{dis}(x, y), \quad y' = y \quad (1)$$

where  $\text{dis}(x, y)$  is the disparity value at the point  $(x, y)$ .

Commonly-used metrics for determining similarity include normalized cross correlation (*NCC*), sum of squared differences (*SSD*), and sum of absolute differences (*SAD*). All of these metrics need a truncation (cut off) value to determine a match between two pixels, i.e., when  $\text{SAD}(A, B) < 100$ , there is a match between pixels  $A$  and  $B$  in the two images. However, one common problem of the above metrics is they need different truncation values for different window sizes. The determination of the truncation value is generally manual and inconvenient.

We use root mean squared error (*RMSE*) as the matching metric

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N [R_i(x, y) - M_i(x', y)]^2} \quad (2)$$

where  $N$  is the total number of pixels in a window,  $R_i(x, y)$  and  $M_i(x', y)$  are intensity values of pixels in the window of the reference image and matching image. In the example of a pair of windows of size 3 in Table 4.1, we have the *RMSE* value of 3.68. The advantage of using *RMSE* is that we can use a universal threshold value for different window sizes to determine a match or non-match without trying and choosing different truncation values.



55	53	55
55	52	56
58	54	58

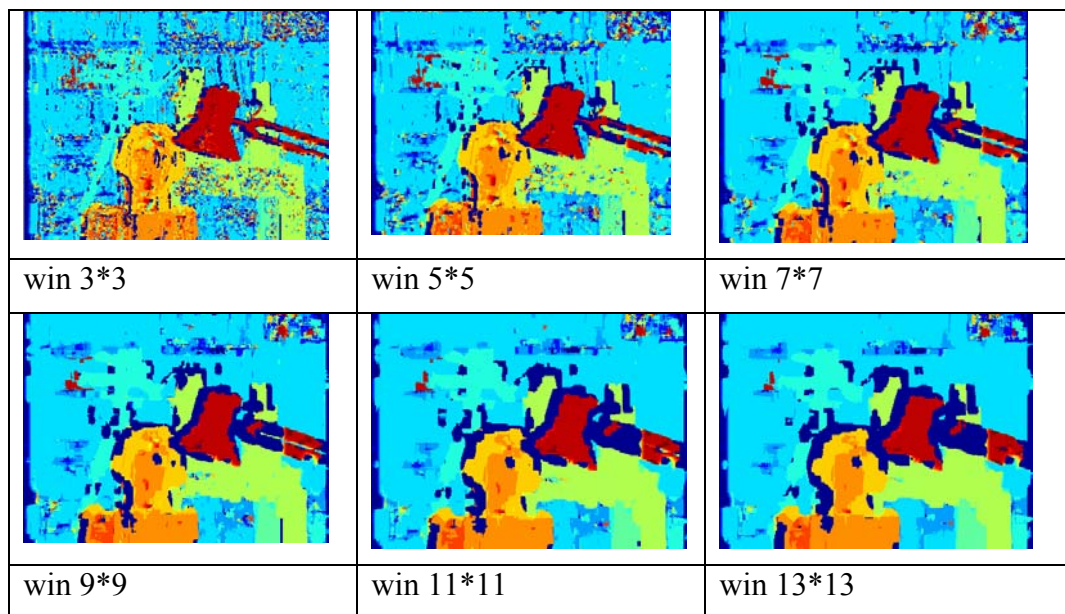
57	56	54
56	54	54
55	56	49

Table 4.1 An example of calculating a RMSE value between two pixels based on the intensity values of the square windows of size 3

For each pixel in each scanline in the reference image, we seek the most similar pixel in the same scanline of the matching image, in terms of the minimum (winning) *RMSE*. If this *RMSE* value is smaller than a threshold value, we conclude that there is a match between the pixels and then calculate their difference along the horizontal axis as the disparity value,  $dis(x,y)=x'-x$ , from Equation 1. Otherwise, we report there is an occlusion or no match here. A disparity map has the disparity values for every pixel in the reference image.

In order to minimize false matches, some matching constraints must be enforced. Besides similarity calculation based on intensity values, other constraints include uniqueness, continuity, ordering, and epipolar constraints [11][12]. Our local stereo method addresses these constraints by default.

Figure 4.16 is the disparity maps from the simple local stereo matching (window based), with window sizes from 3\*3 to 21\*21, and the last map is the ground truth. We can find that window based stereo matching is not satisfactory in both big window matching and small window matching.



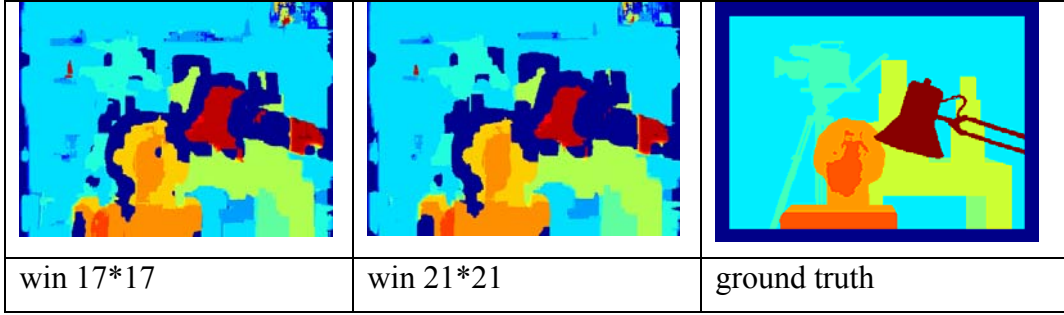


Figure 4.16 Local stereo matching on the stereo data Tsukuba with different window sizes

#### 4.5.2.2 Global Stereo Matching

##### 4.5.2.2.1 Belief Propagation

One representative global stereo matching method is belief propagation [1].

With belief propagation, the stereo matching problem is formulated as a Markov network and solved using Bayesian belief propagation. The stereo Markov network consists of three coupled Markov random fields that model a smooth field for depth/disparity, a line process for depth discontinuity, and a binary process for occlusion. After eliminating the line process and the binary process by introducing two robust functions, the belief propagation is applied to obtain the maximum a posterior (MAP) estimation in the Markov network.

A Markov network is an undirected graph illustrated in Figure 4.17, in which nodes  $\{x_s\}$  are hidden variables and nodes  $\{y_s\}$  are observed variables. Denoting  $X=\{x_s\}$ ,  $Y=\{y_s\}$ , we have

$$P(X | Y) \propto \prod_s \psi_s(x_s, y_s) \prod_s \prod_{t \in N(s)} \psi_{st}(x_s, x_t) \quad (3)$$

where  $\psi_{st}(x_s, x_t)$  is the compatibility matrix between  $x_s$  and  $x_t$ , and  $\psi_s(x_s, y_s)$  is the local evidence, or the observation probability  $p(y_s|x_s)$ .

$$\psi_{st}(x_s, x_t) = \exp(-\rho_p(x_s, x_t))$$

$$\psi_s(x_s, y_s) \propto \exp(-\rho_d(x_s)) \quad (4)$$

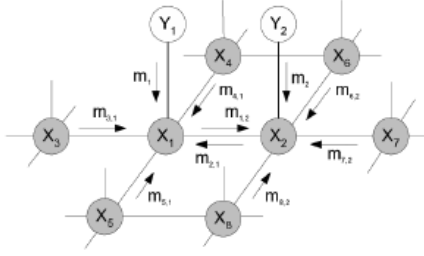


Figure 4.17 Local message passing in a Markov network (gray nodes are hidden variables and white are observed ones)

Belief propagation (*BP*) is an iterative inference algorithm that propagates belief message in the network. There are two kinds of BP algorithms with different message update rules, *max-product* and *sum-product*, which maximizes the joint posterior  $P(X|Y)$  of the network and the marginal posterior of each node  $P(x_s|Y)$  respectively. The steps of *max-product* are:

1. Initialize all belief messages  $m_{st}(x_t)$  as the uniform distributions and  $m_s(x_s) = \psi_s(x_s, y_s)$ .
2. Update belief messages  $m_{st}(x_t)$  iteratively for  $i=1$  to  $T$

$$m_{st}^{i+1}(x_t) \leftarrow \kappa \max_{x_s} \psi_{st}(x_s, x_t) m_s^i(x_s) \prod_{x_k \in N(x_s) \setminus x_t} m_{ks}^i(x_s). \quad (5)$$

3. Compute beliefs

$$b_s(x_s) \leftarrow \kappa m_s(x_s) \prod_{x_k \in N(x_s)} m_{ks}(x_s) \\ x_s^{MAP} = \arg \max_{x_k} b_s(x_k). \quad (6)$$

Belief propagation is widely used and cited in stereo correspondence [6].

#### 4.5.2.2.2 Graph Cut

Graph cut is another commonly-used global stereo correspondence method. Let  $G = \langle V, \mathcal{E} \rangle$  be a weighted graph with two distinguished terminal vertices  $\{s, t\}$  called source and sink. A cut  $C = V^s, V^t$  is a partition of the vertices into two sets such that  $s \in V^s, t \in V^t$ . The cost of the cut  $|C|$  equals the sum of the weights of the edges between a vertex in  $V^s$  and one in  $V^t$  [2]. The minimum cut problem is to find the cut with the smallest cost. The cost (energy) function is defined:

$$E(f) = E_{data}(f) + E_{occ}(f) + E_{smooth}(f) \quad (7)$$

$E_{data}$  results from the differences in intensity between corresponding pixels,  $E_{occ}$  imposes a penalty for making a pixel occluded and  $E_{smooth}$  makes neighboring pixels in the same image tend to have similar disparities.

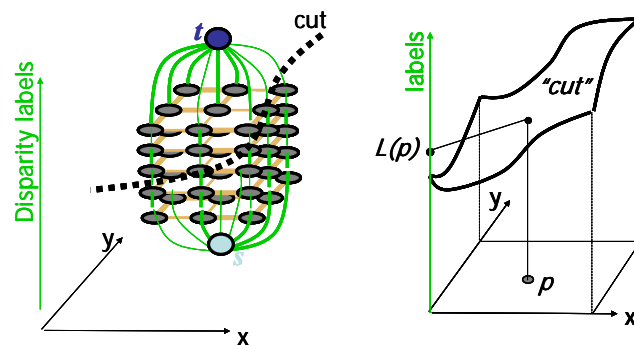


Figure 4.18 Example of finding minimum cost cut that separates source and target

Fast Approximate Energy Minimization via Graph Cuts was proposed to address the problem of minimizing a large class of energy functions that occur in early vision [3]. The major restriction is that the energy function's smoothness term must only involve pairs of pixels. Two algorithms were proposed by *Boykov et al* [3] that use graph cuts to compute a local minimum even when very large moves are allowed. The first move is an  $\alpha$ - $\beta$ -swap: for a pair of labels  $\alpha$  and  $\beta$ , this move exchanges the labels between an arbitrary set of pixels labeled  $\alpha$  and another arbitrary set labeled  $\beta$ . The first algorithm generates a labeling such that there is no swap move that decreases the energy. The second move is an  $\alpha$ -expansion: for a label  $\alpha$ , this move assigns an arbitrary set of pixels the label  $\alpha$ . The second algorithm, which requires the smoothness term to be a metric, generates a labeling such that there is no expansion move that decreases the energy. Moreover, this solution is within a known factor of the global minimum. Experiments demonstrate the approach is effective for image restoration, stereo and motion.



(a) initial labeling      (b)  $\alpha$ - $\beta$ -swap      (c)  $\alpha$ -expansion

Figure 4.19 Alpha-beta-swap and alpha-expansion of graph cut

#### 4.5.2.2.3 Other Stereo Matching Algorithms

Other representative stereo matching algorithms include dynamic programming, symmetric stereo matching, segment-based matching, region-based matching etc. Table 4.2 lists an overview of the stereo matching algorithms with top performance on the Middlebury stereo data [7]. Figure 4.20 has the disparity maps on the data Tsukuba of some of the algorithms according to the Middlebury stereo test bed [6].

Symmetric Stereo Matching for Occlusion Handling (CVPR'05)	visibility constraint, energy minimization, iterative optimization algorithm, belief propagation, segmentation as a soft constraint
A Symmetric Patch-Based Correspondence Model for Occlusion Handling (ICCV'05)	segment-based style, patch-based stereo algorithm, symmetric graph-cuts optimization framework, superior performance on occlusions, untextured areas and discontinuities
Segment-based stereo matching using graph cuts (CVPR'04)	segment-based stereo, graph cuts, energy minimization problem, fast approximate the optimal solution, assigns the corresponding disparity plane to each segment
Graph-based surface reconstruction from stereo pairs using image segmentation	colour segmentation, propagating disparity information to occluded regions, disparity segments clustered, graph cut to optimized cost, pixel level measures the data similarity, segment level propagates the segmentation information
A layered stereo algorithm using image segmentation and global visibility constraints	Colour segmentation, layered model, extracted by clustering of depth planes, Z-buffering enforces visibility & detection of occlusions, greedy algorithm searching for minimum cost func., layer extraction and assignment applied
Stereo matching using belief propagation (PAMI03, ECCV02)	Markov network, using Bayesian belief propagation, a smooth field for depth/disparity, a line process for depth discontinuity, and a binary process for occlusion, MAP, image segmentation
Surfaces with occlusions from layered stereo (Stanford Univ.)	estimates scene structure as a collection of smooth surface patches, Disparities estimated by surface fitting and graph cuts, respectively, energy minimization,
A dense stereo matching using two-pass dynamic programming with generalized ground control points (CVPR'05)	generalized ground control points (GGCPs) scheme, two-pass dynamic programming technique, guarantee to provide sufficient starting pixels needed for guiding the subsequent matching process, reduce the risk of false match
Region-based progressive stereo matching (CVPR'04)	combines the strengths of region-based and progressive approaches. A growing-like process matches the regions progressively using a global best-first strategy based on a cost function integrating disparity smoothness and visibility constraint.
Multi-camera Scene Reconstruction via Graph Cuts (ECCV'02)	Multi-camera scene reconstruction, energy minimization via graph cuts, handles visibility properly, and imposes spatial smoothness while preserving discontinuities, graph cut algorithm computes a local minimum strongly

Table 4.2 An overview of stereo matching algorithms with top performance on the Middlebury stereo data

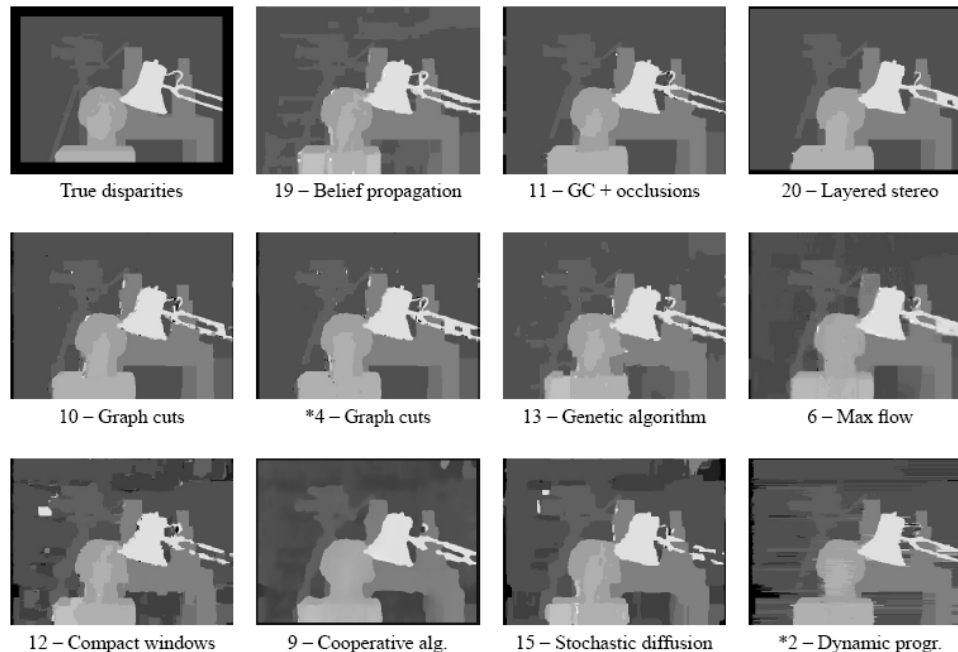


Figure 4.20 The resulting disparity maps of some stereo matching algorithms on the Tsukuba data



#### 4.5.2.3 Evaluations

To evaluate the performance of a stereo algorithm, we need a quantitative estimation of the quality of the stereo correspondence algorithms. A commonly-used approach is to compute the error rate with respect to some ground truth of the disparity maps [6].

$$B = \frac{1}{N} \sum_{(x,y)} (|d_C(x,y) - d_T(x,y)| > \delta_d) \quad (8)$$

where  $B$  is the bad pixel percentage,  $N$  is the total number of pixels,  $d_C(x, y)$  is the computed disparity map and  $d_T(x, y)$  is the ground truth map,  $\delta_d$  is a disparity error threshold.

In addition to computing these statistics over the whole image (all regions), new evaluations on the Middlebury stereo test bed [7] also include the evaluations on non-occluded regions (regions that are not occluded in the matching image) and depth (disparity) discontinuity regions (whose neighboring disparities differ by more than  $eval\_dis\_gap$ , dilated by a window of width  $eval\_discont\_width$ ).

Figure 4.21 is the new evaluation stereo data on the Middlebury stereo test bed [7], and Figure 4.22 has an example of the regions for evaluations on the data *Teddy*, including all regions, non-occluded regions and disparity discontinuity regions.

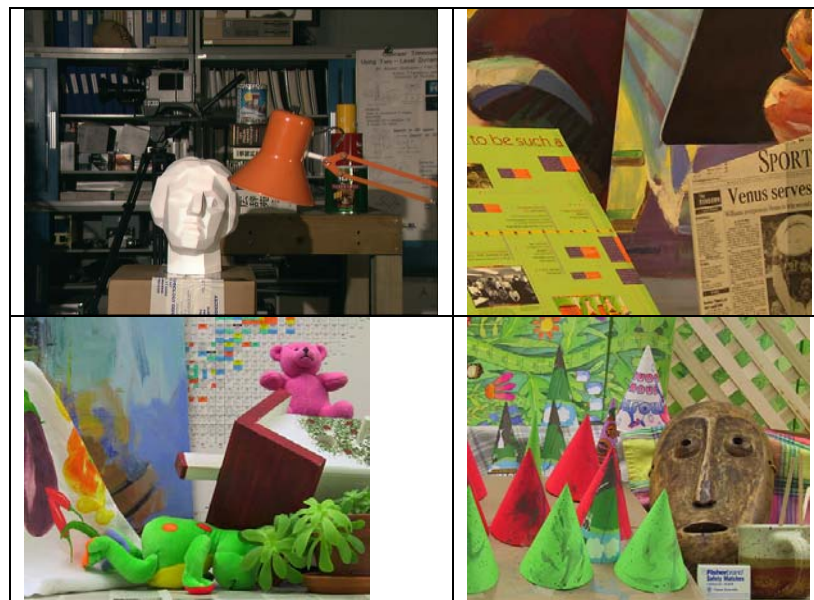


Figure 4.21 The Middlebury stereo data, Tsukuba, Venus, Teddy, and Cones

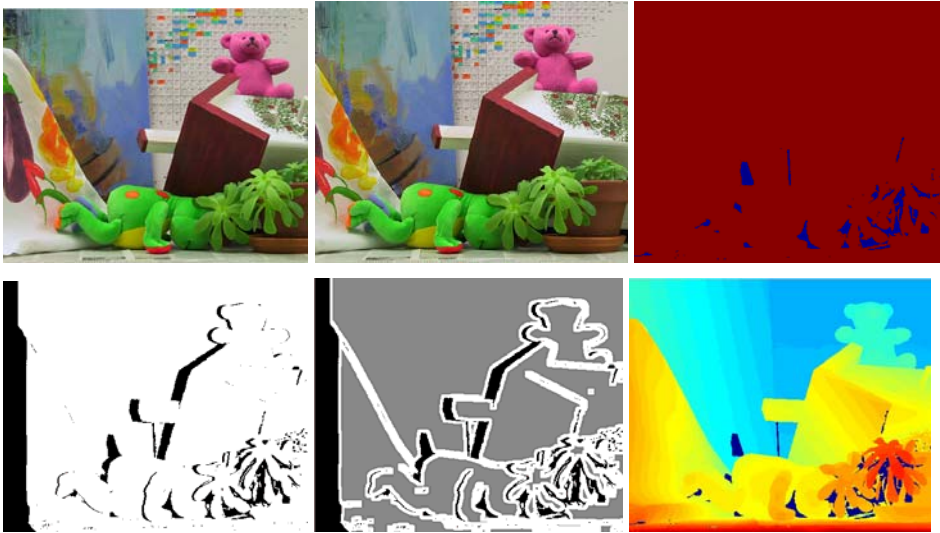


Figure 4.22 Evaluation on the Middlebury stereo data Teddy: (top left) left image, (top middle) right image, (top right) all regions, (bottom left) non-occluded regions, (bottom middle) disparity discontinuity regions, and (bottom right) ground truth

### 4.5.3 Framework

The main idea of our edge-based stereo matching is to progressively integrate the big-window matching and small-window matching using the edges of a disparity map from the big-window stereo matching, so that we can match the densely-textured and textureless regions at the same time. An arbitrarily-shaped windows matching is used for the regions where a regular small window matching and big window matching can not make stereo matches. Instead of using energy-minimization based optimization methods, we propose an optimization method called *Progressive Outlier Remover (POR)* to optimize the disparity maps. This disparity unifying process for differently-sized windows can be repeated until the small window is a window of size 3 or an arbitrarily-shaped window.

#### 4.5.3.1. Arbitrarily-shaped windows

We propose an arbitrarily-shaped window based stereo matching to accurately capture disparity values for densely-textured regions. As illustrated in Figure 4.23, where a regular square window can not find matches for certain regions in the pair of images (see Figure 4.23(a)), an arbitrarily-shaped window can do it (see Figure 4.23(b)). Our arbitrarily-shaped-window strategy is to try out all kinds of shapes and orientations and pick the winning shape that has the minimum similarity value in terms of *RMSE*.

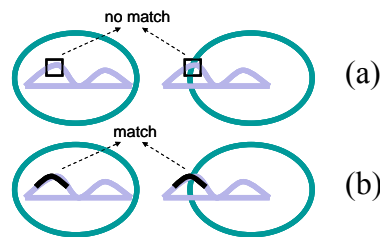


Figure 4.23 An illustration of using arbitrarily-shaped windows (a) a regular square window can not find matched for certain regions, and (b) an arbitrarily-shaped window can

The arbitrary shapes or orientations come from two scenarios, scenario A and scenario B. Each shape or orientation is actually a unique combination of 5 neighboring pixels inside a 5\*5 window with the pixel (0, 0) in the middle, which is the active matching pixel.

In scenario A (Figure 4.24(a)), when the first three pixels are (0, -2), (0, -1) and (0, 0), and our searching route for other pixels to form a unique 5-pixel combination ends at one of the other peripheral points, we will have seven different shapes or orientations. Next, starting from another peripheral pixel and ending at a different peripheral one, we will have six (excluding the shape/orientation found in the previous search). Continue this searching until every peripheral starting pixel is tried, we will have a total of  $\sum_{i=1}^7 i = 28$  shapes/orientations for scenario A. For example, the horizontal window across the point (0, 0) can be represented as (0, -2), (0, -1), (0, 0), (0, 1), (0, 2), where the x, y values of the points are the vertical and horizontal differences from the active matching pixel (0, 0).



In scenario B (Figure 4.24(b)), we use the remaining peripheral pixels of the 5\*5 square from scenario A when forming the unique 5-pixel combinations. When our first three pixels are (-1, -2), (0, -1) and (0, 0), we will have 15 different shapes or orientations. Taking other searching routes to form unique 5-pixel combinations, and keeping (0, 0) as the central pixel and start point and end point peripheral pixels of the square, we will get  $\sum_{i=1}^{15} i = 120$  unique shapes or orientations.

For the highlighted example of Figure 4.24(b), the window is represented as (-1, -2), (0, -1), (0, 0), (0, 1), (-1, 2). Summed from these two scenarios, we will have a total of 148 different shapes/orientations to pick a 5-pixel arbitrarily-shaped window.

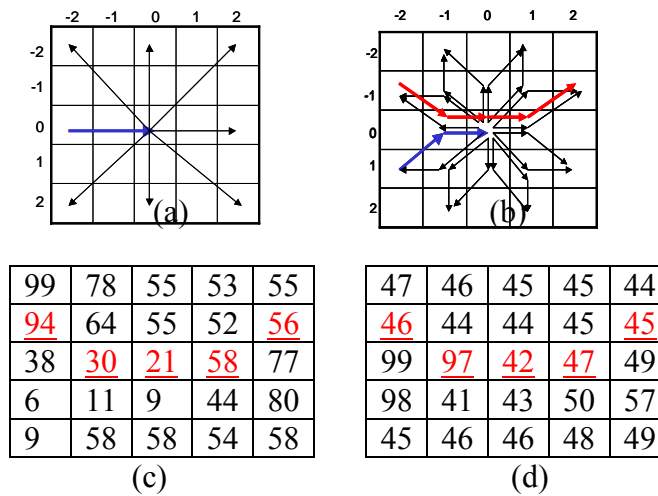


Figure 4.24 Arbitrarily-shaped windows (a) scenario A, (b) scenario B, (c) (d) matching pixels in the pair of images based on the highlighted example window in (b)

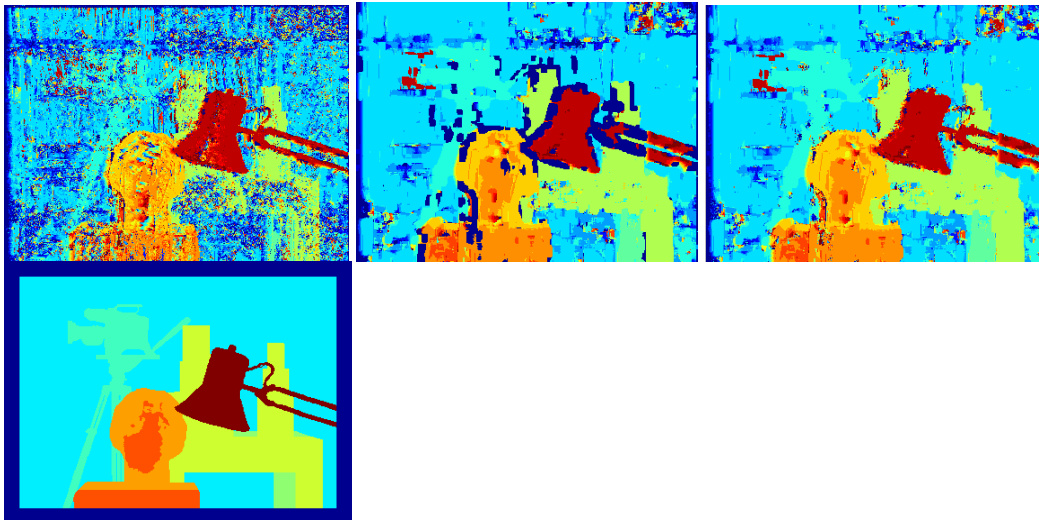


Figure 4.25 Resulting images (top left) arbi-shaped window matching, (top middle) window 7\*7 matching (top right) window 7\*7 + arbi-shaped window matching, and (bottom) ground truth

In Figure 4.25, with the help of arbitrarily-shaped window matching, a  $7 \times 7$  window matching has improved stereo correspondence performance. However, we need further optimization.

We use five as the pixel number of an arbitrarily-shaped window, because three will be too small to compute reliable matching costs and seven and more will be cost prohibitive. Comparing with regular square windows, the computation time for an arbitrarily-shaped window based matching is the single 5-pixel matching time multiplied by 148 ( $5 \times 148$ ), which is equivalent to matching with a square window of size 28 ( $28 \times 28$ ). By applying the arbitrarily-shaped windows only for the regions when a regular window based matching can not find matches (less than 10%), the complexity is greatly reduced.

#### 4.5.3.2. Progressive edge-based stereo matching

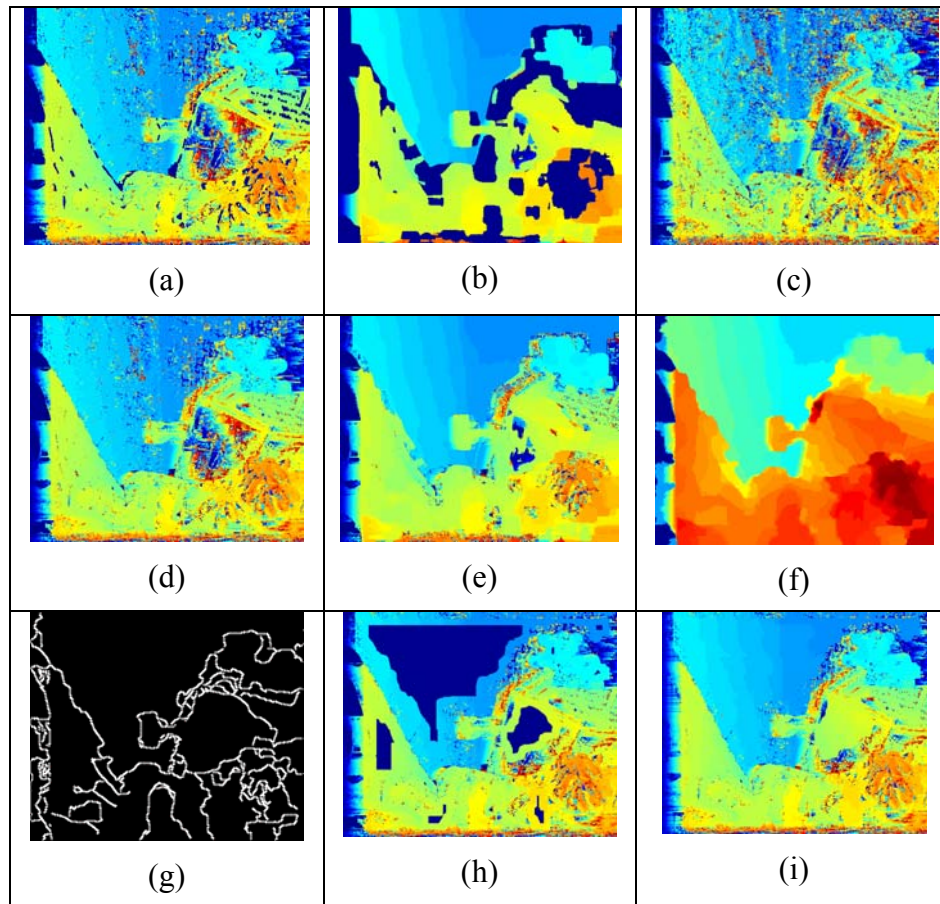


Figure 4.26 Framework illustration of our edge-based stereo correspondence (a)(b)(c) left, center, and right of the top row; (d)(e)(f) left, center, and right of the middle row; (g)(h)(i) left, center, and right of the bottom row

The main steps of our progressive edge-based stereo matching are illustrated in Figure 4.26. As an example of our stereo matching on the image data *Teddy*, Figure 11(a) is the

disparity map from a small-window matching (briefly *win\_small*, of the size 3\*3, or *W3*); (b) is the disparity map from a big window matching (*win\_big*, of the size 25\*25, or *W25*); (c) is the disparity map from the arbitrarily-shaped windows matching (*win\_arbi*, or *Wa*); (d) is *W3+Wa*; (e) is *W25+Wa*; (f) is the optimized *W25+Wa* by the *POR*; (g) is the edges of the disparity map of (f); (h) is the disparity strips combined from (d) and (e) according to the edges in (g); and (i) is the smoothed disparity map from (h), which will be further optimized by the *POR* optimization method, and serve as the big window disparity map for the next round of edge-based stereo matching, if necessary.

An optimal edge detector *Canny* [13] is used to detect edges. We use the command *edge\_name = edge(image, 'canny', k)* in MATLAB to get the edge file for the input disparity map image, and *k* is the parameter for *Canny*. A smaller *k* value will generate more edges in the binary output edge file, in which 1 represents an edge and 0 otherwise.

When combining the big window and small window matching, with arbitrarily-shaped windows used for regions where a big window or small window can not make matches, we use the disparity values from *win\_small+win\_arbi* for the strips around the edges; and use the disparities from *win\_big+win\_arbi* for the strips away from the edges (next to the *win\_small+win\_arbi* strips). The width of *win\_small+win\_arbi* strips at each side of the edges and that of the neighboring *win\_big+win\_arbi* strips are

$$W_{strip} = \frac{1}{2}[size(win\_big) - size(win\_small)] + 1 \quad (9)$$

We enforce the disparity continuity between the strips of *win\_big+win\_arbi* using a disparity averaging scheme. Suppose a pixel (*x*, *y*) inside the region is to be smoothed, the disparity value *dis*(*x*, *y*) depends on the closest disparity values of four directions on its neighboring strips (horizontally left and right disparities *dis*(*x*<sub>1</sub>, *y*) and *dis*(*x*<sub>2</sub>, *y*), and vertically above and below disparities *dis*(*x*, *y*<sub>1</sub>) and *dis*(*x*, *y*<sub>2</sub>)), then

$$\begin{aligned} dis_x(x, y) &= \frac{dis(x_2, y) - dis(x_1, y)}{x_2 - x_1}(x - x_1) + dis(x_1, y) \\ dis_y(x, y) &= \frac{dis(x, y_2) - dis(x, y_1)}{y_2 - y_1}(y - y_1) + dis(x, y_1) \\ dis(x, y) &= \frac{1}{2}[dis_x(x, y) + dis_y(x, y)] \end{aligned} \quad (10)$$

#### 4.5.3.3. The progressive outlier remover optimization

Our Progressive Outlier Remover (POR) optimization algorithm is based on the disparity continuity assumption: in a small region, when a disparity value is greatly different from its surroundings, it is deemed as an outlier and should be replaced or optimized.

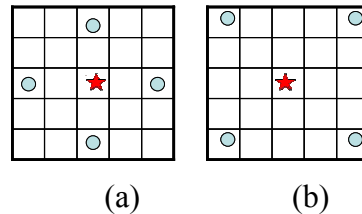


Figure 4.27 POR optimization method: an outlier disparity is replaced with an average of its four neighbors' disparities in either case of (a) and (b)

Illustrated in Figure 4.27, for each value in the disparity map, we compare it with four equally-distanced neighbors in four directions, separately, one kind of neighbors are directly above, below, left, and right neighbors (Figure 4.27(a)), and another kind are four corners of a square where the current pixel is centered (Figure 4.27(b)).

When the disparity of the central pixel is not equal to any of its neighbors' disparities, and its difference from the average of the neighbors' disparities is bigger than a threshold, it will be replaced by the average of the neighbors' disparities. The threshold  $T$  is proportional to the product of the neighbors' vertical or horizontal distance  $d$  to the central pixel and the standard deviation  $\sigma$  of the four neighbors' disparity values.

$$T = k \cdot d \cdot \sigma \quad (11)$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2 - \left(\frac{1}{N} \sum_{i=1}^N x_i\right)^2}$$

where  $N = 4$ , and  $k$  takes a value of 1 or 0.5. A  $k$  value of 1 represents relatively stricter thresholds than that of 0.5.

The *POR* optimization algorithm takes two parameters, distance  $d$  (a value usually from 2 to 20), and a decremental rate  $R$  (a value of 2/3, 1/2 or 2/5) for getting decreasing iteration numbers in different rounds of iterations. For each round  $i$ , we use the iteration number

$$n_i = \lfloor d \times R^{i-1} - 0.5 \rfloor + 1 \quad (12)$$

For example, when we have  $(d, R) = (20, 2/3)$ , we will have 8 rounds of optimizations with  $n_i = \{20, 13, 9, 6, 4, 3, 2, 1\}$ ; for  $(d, R) = (20, 1/2)$ ,  $n_i = \{20, 10, 5, 3, 2, 1\}$ ; and for  $(d, R) = (20, 2/5)$ ,  $n_i = \{20, 8, 3, 1\}$ . For each round of iterations, we have 1 and 0.5 as the  $k$  values alternatively (Equation 11), i.e., we have (20, 1), (20, 0.5), (8, 1), (8, 0.5), (3, 1), (3, 0.5), and (1, 1) as the  $(n_i, k)$  combinations for  $(d, R) = (20, 2/5)$  (note we do not have (1, 0.5)). For each round, with the iteration number  $n_i$ , the *POR* algorithm will have iteration  $i$  from 1 to  $n_i$ , each of which has the distance (from the central pixel to any of its neighbors) of  $i$ , and have the threshold for outlier removal of  $k \cdot i \cdot \sigma_i$  defined in Equation 11. For reliability purposes, we repeat each step of iteration  $i$  twice.

A complete *POR* algorithm is in Figure 13.

---

*Algorithm:* Go-light ( $d, R, \text{dis}(\text{disparity map})$ )

For round  $i=1, i++, n_i = \lfloor d \times R^{i-1} - 0.5 \rfloor + 1$ , until  $n_i \leq 1$

For  $k\_round=1:2$

    If ( $k\_round==1$ )  $k=1$ ; else  $k=0.5$ ;

    For  $n=1; n \leq n_i; n++$

        distance  $d=n$ ,

threshold  $T_i = k * d * \sigma_i$  (Equation 11,  $i=A, B$  scenarios (Figure 12))

For each point  $dis(x, y)$  in the disparity map,

If:  $dis(x, y) \neq \forall \in \{dis(x-d, y), dis(x+d, y), dis(x, y-d), dis(x, y+d)\}$  &&  $abs(dis(x, y) - average_A) > T_A$  (here ' $\forall \in$ ' means 'any of')

Then:  $dis(x, y) = average_A(dis(x-d, y), dis(x+d, y), dis(x, y-d), dis(x, y+d))$

If:  $dis(x, y) \neq \forall \in \{dis(x-d, y-d), dis(x+d, y+d), dis(x+d, y-d), dis(x-d, y+d)\}$  &&  $abs(dis(x, y) - average_B) > T_B$

Then:  $dis(x, y) = average_B(dis(x-d, y-d), dis(x+d, y+d), dis(x+d, y-d), dis(x-d, y+d))$

Figure 4.28 The Progressive Outlier Remover (POR) optimization algorithm

An example of applying this algorithm on the stereo image *Venus* is in Figure 4.29.

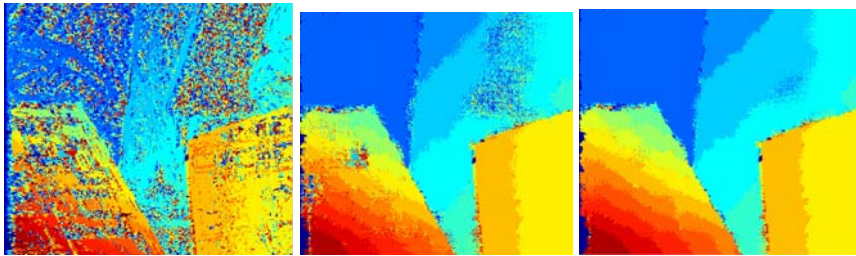


Figure 4.29 Applying the POR (a) disparity map of W3Wa (win\_size3 +win\_arbi) of the image *Venus*, (b) (c) after the first two rounds of POR optimizations with  $d=14$ ,  $R=1/2$

#### 4.5.4 Experimental Design and Results

We work on four *Middlebury* stereo images, *Tsukuba*, *Venus*, *Teddy*, and *Cones* for quantitative evaluation, which are the benchmark data for stereo correspondence algorithms [7]. We evaluate our algorithm in terms of the percentage of bad pixels, i.e., pixels whose absolute disparity error is greater than a threshold (such as 1 and 0.5). We calculate percentages for (1) pixels in non-occluded regions, (2) all pixels and (3) pixels near disparity discontinuities, and ignore a border of 10 pixels for *Venus*, and 18 for *Tsukuba* when computing statistics, according to the evaluation standard on the *Middlebury* stereo [6].

For window based stereo matching, we use *RMSE* as the cost metric, and use 15 as a universal cut off value for determining a correspondence, based on our preliminary empirical experiments.

We firstly experiment on arbitrarily-shaped window matching using the POR optimization, without using the edge-based strategy. We design different extents of using the arbitrarily-shaped windows: a pure 3\*3 window based matching (no arbitrarily-shaped windows, we call this case 1), a 90%-square-window-matching plus a 10%-arbitrarily-shaped-window-matching (case 2), and a 10%-square-window-matching plus a

90%-arbitrarily-shaped-window-matching (case 3). For case 2, we use the arbitrarily-shaped windows for locations where a regular 3\*3 window can not find matches. For case 3, we use the regular square window with size of 9, 11 or 13 for the regions that the arbitrarily-shaped windows can not find matches, considering that arbitrarily-shaped windows are good at matching highly textured areas and big windows are good at matching textureless ones. Experimental result shows that except for the data Teddy, using of arbitrarily-shaped windows produce more accurate disparities (Table 4.3), and overall, case 2 performs the best of the three different usages of arbitrarily-shaped windows. It takes case 2 about 8 minutes to match the data Tsukuba, working on an Intel Pentium 4, 1G memory computer (a longer running time than global methods, but reasonable for a local one).

all %	Tsukuba	Venus	Teddy	Cones
case 1	5.21	4.07	<b>22.39</b>	18.36
case 2	<b>4.92</b>	3.57	22.62	<b>17.51</b>
case 3	4.99	<b>3.48</b>	22.66	19.07
nonocc %	Tsukuba	Venus	Teddy	Cones
case 1	3.54	2.98	<b>14.48</b>	10.11
case 2	<b>2.98</b>	2.52	16.49	<b>9.78</b>
case 3	3.22	<b>2.47</b>	16.44	11.68

Table 4.3 The performance of using different extents of arbitrarily-shaped windows (in terms of percentage bad pixels for non-occluded regions and all regions)

With the progressive edge-based stereo matching, our big window and small window matching are actually  $win\_big+win\_arbi$  and  $win\_small+win\_arbi$ , where the arbitrarily-shaped windows matching is applied to the regions that a regular big window or small window can not make matches. When optimizing the disparity map using the *POR* optimization algorithm, we try different parameters of  $d$  and  $R$  and pick the disparity maps with the best accuracy.

	Canny (k)	Progressive edge-based stereo matching steps	Final POR (d,R)
Tsukuba	0.05	W5WaP3+W3WaP3+Wa	(4, 2/5)
Venus	0.05	W21WaP3+W9WaP3+W3Wa	(14, 2/5)
Teddy	0.05	W25WaP12+W3Wa	(6, 2/3)
Cones	0.001	W9WaP3+W5WaP3+W3Wa	(11, 2/5)

Table 4.4 The parameter settings for the four stereo data

Table 4.4 lists the parameters we used for each of the stereo data, in which  $W5WaP3$

means window size 5 + arbitrarily-shaped windows, optimized by *POR* with  $d=3$ , *Wa* means arbitrary-shaped windows without *POR* optimization, and similar for others. A small-scale *POR* optimization (e.g.,  $d=3$ ) can make the disparity map smooth while keeping the global disparity distribution. The number of plus signs ('+') in Table 4.4 is also the number of rounds of the progressive edge-based stereo matching.

The data *Teddy* has a big textureless area, which a regular local stereo algorithm has difficulty to deal with. By using a big window match of size 25, optimized by *POR* of  $d=12$ , the big hole in the textureless regions of the disparity map is then smoothed. For the representative densely-textured data *Tsukuba*, we use relatively small window sizes and small parameters for the *POR* optimization, to avoid the loss of the accurate disparities for the delicate textures. Except for *Teddy*, we progressively use two rounds of edge-based stereo matching for all other stereo data. The output disparity map of the first round edge-based matching is inputted to the second round matching as the big-window disparities.

The overall evaluation of our algorithm is in Table 4.5, Table 4.6 and Figure 4.30. By the time of submission, the average rankings of our algorithm on the *Middlebury* stereo evaluation webpage [7] are No. 22.0 for error threshold of 1, and No. 16.9 for error threshold of 0.5, out of 29 submissions to the system, most of which are results from published state-of-the-art algorithms. Compared with other disparity optimization methods, our algorithm is better than scanline optimization [6] and comparable with graph cuts using  $\alpha$ - $\beta$  swaps [8] and dynamic programming [14], on the new version of *Middlebury* evaluation. On the previous version of *Middlebury* evaluation data, our algorithm is better than other window-based stereo correspondence algorithms such as the pixel-to-pixel algorithm [9] and comparable with the discontinuity preserving algorithm [10] and the variable window algorithm [4].

With the threshold of 0.5, our progressive edge-based stereo matching has the average rankings of No. 14 for the non-occluded regions and all regions apiece, but has the average ranking of No. 22 for the disparity discontinuity regions. We plan to improve this algorithm in our future work, especially for its performance of matching the disparity discontinuity regions.

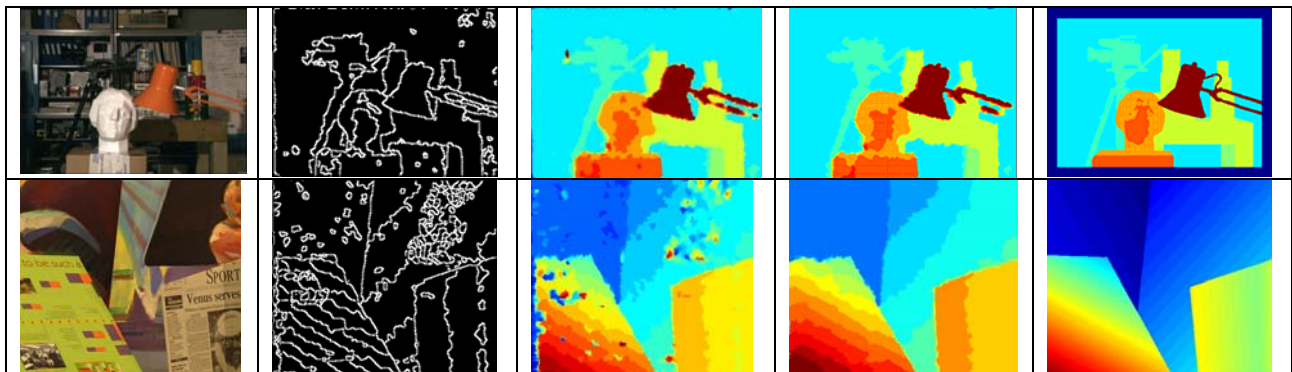


	Tsukuba			Venus		
	nonocc	All	disc	nonocc	all	disc
w/o edge-based	2.98	4.92	15.1	2.47	3.48	27.5
edge-based	<b>2.73</b>	<b>4.65</b>	<b>13.9</b>	<b>2.25</b>	<b>3.24</b>	<b>27.4</b>
	Teddy			Cones		
	nonocc	all	disc	nonocc	all	disc
w/o edge-based	14.5	<b>22.4</b>	33.0	9.78	17.5	21.3
edge-based	<b>14.3</b>	23.1	<b>30.2</b>	<b>7.63</b>	<b>16.1</b>	<b>19.7</b>

Table 4.5 Improvement of using progressive edge-based stereo matching over without using edge-based strategy (in terms of percentage of bad pixels for non-occluded, all and disparity discontinuity regions, with threshold of 1)

	Tsukuba			Venus		
	nonocc	All	disc	nonocc	all	disc
Thre1	2.73 <sub>19</sub>	4.65 <sub>20</sub>	13.9 <sub>24</sub>	2.25 <sub>21</sub>	3.24 <sub>21</sub>	27.4 <sub>27</sub>
Thre0.5	8.26 <sub>5</sub>	10.4 <sub>6</sub>	23.0 <sub>21</sub>	8.57 <sub>13</sub>	9.67 <sub>13</sub>	33.9 <sub>25</sub>
	Teddy			Cones		
	nonocc	all	disc	nonocc	all	disc
Thre1	14.3 <sub>22</sub>	23.1 <sub>22</sub>	30.2 <sub>26</sub>	7.63 <sub>20</sub>	16.1 <sub>20</sub>	19.7 <sub>22</sub>
Thre0.5	24.3 <sub>20</sub>	32.2 <sub>22</sub>	43.0 <sub>24</sub>	15.0 <sub>17</sub>	23.0 <sub>17</sub>	28.0 <sub>20</sub>

Table 4.6 An evaluation of our algorithm on the Middlebury data (in terms of percentage of bad pixels for non-occluded, all, and disparity discontinuity regions; the subscripts of the results are our rankings amongst other state-of-the-art algorithms, with thresholds 1 and 0.5)







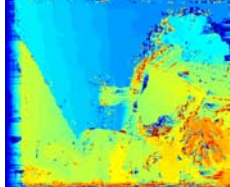
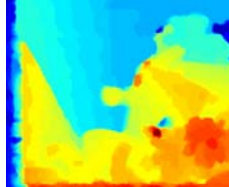
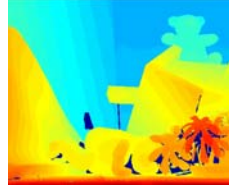


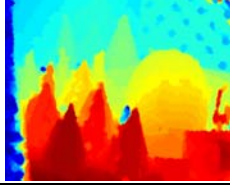
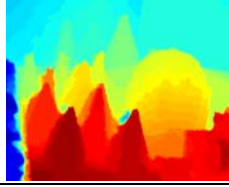
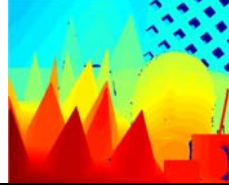
				
				
Reference image	Edges of win_big + win_arbi (first round)	win_big and win_small combined (first round)	Disparity map after the final POR optimization	Ground truth disparity maps

Figure 4.30 The results of our progressive edge-based stereo matching algorithm (from top to down: Tsukuba, Venus, Teddy, and Cones. Same color on different maps does not necessarily represent the same disparity)

#### References for Section 4.5

- [1] J. Sun, N-N. Zheng, and H-Y. Shum, Stereo Matching Using Belief Propagation, *PAMI*, 25(7), 2003.
- [2] V. Kolmogorov and R. Zabih, Computing Visual Correspondence with Occlusions using Graph Cuts, *ICCV*, 2001.
- [3] Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, *ICCV*, 1999.
- [4] O. Veksler, Fast Variable Window for stereo correspondence Using Integral Images, *CVPR*, 2003.
- [5] J.C. Kim, K.M. Lee, B.T. Choi, and S.U. Lee, A Dense Stereo Matching Using Two-Pass Dynamic Programming with Generalized Ground Control Points, *CVPR*, 2005.
- [6] D. Scharstein and R. Szeliski, A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms, *IJCV*, 47(1), pp. 7-42, 2002.
- [7] Middlebury stereo, <http://www.middlebury.edu/stereo>
- [8] Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, *PAMI*, 23(11), 2001.
- [9] S. Birchfield, and C. Tomasi, Depth Discontinuities by Pixel-to-Pixel Stereo, *ICCV*, 1998.
- [10] M. Agrawal, and L. Davis, Window-Based Discontinuity Preserving Stereo, *CVPR*, 2004.
- [11] D.C. Marr, and T. Poggio, A Computational Theory of Human Stereo Vision, *Proceedings of the Royal Society of London*, B(204), pp.301-328, 1979.
- [12] H. Baker, and T. Binford, Depth from Edge and Intensity Based Stereo, *IJCAI*, pp.631-636, 1981.
- [13] J. Canny, A Computational Approach To Edge Detection, *PAMI*, 8, pp. 679-714, 1986.
- [14] A.F. Bobick, and S.S. Intille, Large Occlusion Stereo, *IJCV*, 33(3), pp.181-200, 1999.

## 4.6 3D and Multi-View Video Technologies

This part of the project relates to developing a 3D Video player and algorithms for 3D video compression. The use of 3D video improves the surveillance and monitoring capabilities. A pair of cameras is used to capture 3D video. Cameras developed by Dr. Bill Glenn's group capture quad-HD resolution video and take up large amount of storage and communication resources. An array of such cameras can be used to create multi-view video and to monitor large areas. New algorithms are necessary to process large amounts of data and to exploit the correlation among the multiple views. In this project we developed a 3D Video Player, algorithms for 3D video compression, and algorithms for multi-view video compression.

### 4.6.1 Introduction

The recent interest in 3D and multi-viewpoint (MV) TV can be attributed, in part, to the success of the MPEG-4 AVC/H.264 video coding standard. The coding gains made possible by H.264 can be applied to provide enhanced services such as multi-viewpoint TV and 3D television. Another reason for the increasing interest in 3D TV is the recent advances in the display technologies that have lowered the cost of stereoscopic projectors and 3D displays. While these technological advances have renewed interest in 3D/multi-view coding, the successful deployment of 3D services still faces key challenges. The current state of the technology and the maturity of the marketplace indicated that this is the right time to overcome barriers to 3D and MV TV services.

The digital video revolution launched by the MPEG-1 and MPEG-2 video coding standards also resulted in an active 3D and multi-view video coding research [1, 2]. The MPEG-2 multi-view profile is a form of temporal scalability that encodes left view of the stereo pair as a base layer and the right view is coded as a temporal enhancement. Existing studies on the quality of 3D video are based on MPEG-2 view coding and not applicable to H.264 based coding that is expected to be used in 3D TV services [3]. The studies also did not use autostereoscopic displays which are expected to be the dominant display types for 3D TV [4]. MPEG-2 based coding is inefficient compared to H.264 based view coding. Furthermore, the coding artifacts in MPEG-2 and H.264 are different and are likely to have different effects on the 3D perception. The quality of a 3D video experience is influenced by the type of displays used. A good summary of the perceptual quality requirements and evaluations for 3D video is presented in [4]. Our current focus is on developing efficient coding and representation algorithms for 3D and multi-view video. We are using H.264 as the basis for view coding and autostereoscopic displays for rendering the 3D video.

One of the reasons for the lack of success of 3D TV so far is the ease-of-use of the 3D TV and the viewing comfort. Most of the displays today use standard TV with anaglyph video and a pair of glasses to generate 3D perception. Watching such TV is straining to the eye. Even the current generation autostereoscopic displays have limited viewing angle

and are not suitable for viewing for longer periods. The application where 3D video has had reasonable success are the applications where viewing comfort is secondary to the objective; applications such as security, medicine, design automation, and, scientific visualization.

#### 4.6.2 Overview of Multi-View Video System

The 3D and multi-view video coding system was developed with focus on security and surveillance. The goal of this project is to develop technologies and tool for efficient compression, communication, and playback of multi-view and 3D video.

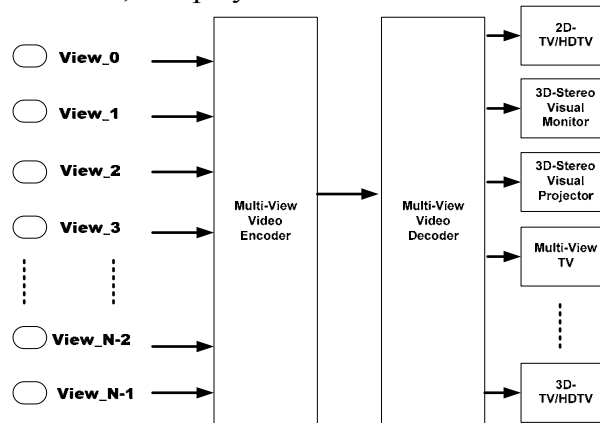


Figure 4.31 3D/Multiview video system

Figure 4.31 shows the general architecture of a multi-view video system. The multiple views are encoded at the sender by exploiting the large amount of redundancies among the views. We use H.264 as the core compression engine with inter-view prediction to increase compression efficiency [5]. The coded views are communicated to the receiver where the decoded views are rendered on an appropriate display. The 3D displays use a pair of coded views to display 3D video with depth perception.

##### 4.6.2.1 Brief Overview of Binocular Vision

The human visual system receives two separate projections of a scene; one from each eye. The eyes are separated by an average horizontal distance of 6.3 cm [7]. The stereoscopic image is an image synthesized by the monocular left-eye-view and the monocular right-eye-view causing relative viewing projections described with high correlation, but with different image information. The left and right eye views are combined resulting in a single 3D percept. The combined visual perception of the scene is also known as binocular fusion. Binocular suppression is property where portions of the view in one eye are suppressed by the corresponding view of the other eye. The possibilities of dominance and suppression mechanisms during the binocular fusion exist, but their impact is not yet well understood [7]. Experiments have shown that when the left and right eye views are combined the higher quality view is able to mask coding artifacts in the lower quality view [3,8].

The process of binocular fusion in the human visual system results in the comparison and combination of the left and right eye views to generate a single 3D percept. The left and right eye views have to be presented to the users using 3D display means to give the sensation of 3D and depth perception. The left and right eye views can be encoded and sent to the receiver and the stereo views can be generated at the receiver. The properties of binocular fusion make possible encoding of left and right eye views at different bitrates. This asymmetric view coding has been exploited to improve compression efficiency [3,8]. The H.264 video coding used in our system is much more efficient than MPEG-2 and also has support for de-blocking that improves the perceptual quality of video. The effects of these improved compression algorithms and autostereoscopic displays on the 3D video quality cannot be understood from the past MPEG-2 based studies.

The two main approaches to delivering 3D video are 1) stereo coding where the left and right views are encoded and 2) depth image based rendering (DIBR) where a single view and an associated depth map are transmitted to the receiver [9]. DIBR systems synthesize the left and right views at the receiver based on the single view and the depth information. These two approaches have their advantages and disadvantages. However, from a production and compatibility point of view the stereo coding methods are more suitable. Furthermore, the free viewpoint TV (FTV) based on multi-view video coding (MVC) is gaining momentum and this makes DBIR approaches unnecessary as the MVC is sufficient to generate the left and right views necessary for the 3DTV.

#### 4.6.3 Stereoscopic and Multi-View Video Player

While the study stereoscopic visual stimuli is not new, it is a field that has seen renewed interest due to advances in capturing videos, mediums for broadcasting, autostereoscopic displays, and other viewing techniques. This section presents the architecture of a modular video player with stereoscopic and multi-view capabilities.

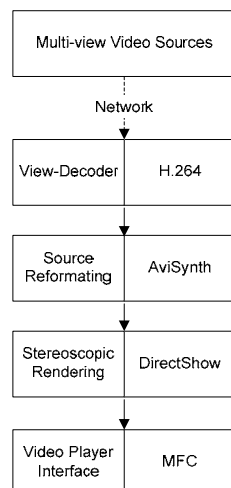


Figure 4.32 3D/MV player architecture

#### 4.6.3.1 Player Architecture

The player was implemented and tested on the Microsoft Windows XP platform. The Microsoft DirectShow framework was used for the capture and transform functions. MFC was used to implement the interface. An open source project, AviSynth [10], was used for some preprocessing tasks. The player takes a pair of views as input and renders them in a format suitable for the target display (anaglyph, Sharp 3D display, side-by-side, etc.). The inputs can be from video decoded from the network or from local video sources (e.g., files, cameras). Figure 4.32 shows this general architecture.

DirectShow is a component of DirectX. DirectShow offers a modular architecture that allows runtime reuse of modules (known as DirectShow filters). The framework allows reusing existing filters for video capture, decoding, and rendering. Filters are connected via compatible terminals, known as pins. A collection of connected filters is referred to as a graph. A minimal graph consists of a source filter to decode media, a transform filter to perform a meaningful operation on the media, and a render filter to display the result on screen or write it to disk. Because our player deals with known and widely available video codecs we are not concerned with source and render filters. Additionally, the use of AviSynth abstracts an even wider variety of file formats that could not normally be played back (for example, raw YUV files) by presenting them as uncompressed AVI data to the player. Instead, the transform filter is where the majority of the processing takes place. In our project the transform filter changes depending on the choice of output format (monoscopic or a specific stereoscopic format).

There are many choices for the implementation of an interface. One option is simply to write a series of DirectShow filters that can be used with a variety of preexisting media players. The existing players lack support for multi-view and 3D sources and player thus needed a new interface. Windows MFC provides as much control over the interface as needed in a Windows environment and is well-documented.

We chose to include AviSynth in our project for several reasons. It is an open source project that has been in use for several years. As a result we trust the validity of its functionality, such as color-space conversions, and can verify the implementation for ourselves. Using AviSynth resulted in considerable time savings, enabling us to focus our work on our primary goal of rendering stereoscopic video.

#### 4.6.3.2 Stereoscopic Video Playback

One of the challenges of displaying stereoscopic video is the wide variety of video formats. Stereoscopic video is typically available as independent left and right sequences or as a single video formatted with the left and right views side-by-side or top-to-bottom. In the implemented solution we use the versatile AviSynth scripting language to help format stereo video data consistently for the stereo player. AviSynth is a frame server. It performs a variety of transformations on video files on-the-fly without creating other files. To the player application the AviSynth script appears as an uncompressed AVI file.

In practice we found AviSynth to provide a useful layer of abstraction between the source data and the player, greatly reducing the complexity of the player.

The user must be able to specify the format of the source video data. For example, if we desire to playback left and right video data encoded in two separate files the AviSynth script needed would ensure that the videos are of equal length and resolution and then place them side-by-side with the left source to the left. This is the format that is expected by the video player. Similar transformations can be made for other formats. If the source is a single video in the side-by-side format no changes are needed. The AviSynth script needed to format the video for playback can be generated with the assistance of a GUI and does not need to be written by the user. The specification of a video format and the generation of the corresponding AviSynth file are performed only once.

### 4.6.4 Experimental Methodology

The goal of this work is to understand the impact of the compression advances in H.264 video and the display advances in the autostereoscopic displays on the quality of the 3D video experiences. We are currently conducting a large user study to evaluate the impact of asymmetrically coded 3D views on the quality of the 3D video rendered on the Sharp autostereoscopic display. The goal of this study is to understand the bounds of asymmetric coding, relationship between the eye-dominance and 3D quality of asymmetrically coded video, and to understand the effects of the H.264 coding features that improve perceptual video quality. The results are reported based on the evaluations from 14 users that have evaluated the subjective quality so far.

The sequences used for these experiments are the Akko & Kayo and the Ballroom sequences created for 3D/multiview coding work currently underway in the MPEG committee [11]. A pair of views from these sequences was chosen to render stereo video. The video sources are 10 seconds long, 640x480 resolution, 30 FPS, and available in YUV 4:2:0 format. The Akko & Kayo sequence is made specifically for this research and has a number of carefully selected objects that help evaluation of 3D sequences well. The Ballroom sequences capture ballroom dancing and show dancers at multiple levels of depth.

The test sequences were created to test 3D video at different levels of quality. The quality was varied by encoding the left and right eye views at different qualities. Two test cases were created for each video sequence: 1) right eye view at a high quality with left eye view quality varying and 2) left eye view kept constant at a high quality and the right eye view quality varying. The high constant quality views were encoded at a PSNR of 42.5 dB, considered broadcast quality, and the quality of the other view is varied from 42.5 dB to 28 dB. The discussion presented here uses PSNR for quality and deliberately avoids using bitrate as there is no standard way of encoding 3D video yet and the same quality can be achieved at different bitrates depending on the coding and prediction modes used. Subjects were recruited to participate in this research and evaluate the 3D viewing experiences. This is an ongoing study and the results reported are for 16 subjects evaluating the test sequences. The participants evaluated the overall quality of video

(without looking for specific artifacts) on the standard subjective evaluation scale from 1 to 5 (1-bad, 2-poor, 3-fair, 4-good, 5-excellent). Most of the participants have had 3D movie experience in the past but this evaluation was the first experience with autostereoscopic displays. Before beginning the evaluations, the participants were shown four high quality 3D video sequences including the two test sequences without any compression.

We used the Sharp LL-151-3D autostereoscopic display to render the stereoscopic videos. The display is 15-inches, XGA resolution (1024 by 768 pixels). This display which uses lenticular imaging techniques and renders depth very accurately gives a true 3D experience. The perception of depth is achieved by a parallax barrier that diverts different patterns of light to the left and right eye. It should be noted that our player architecture accommodates a variety of formats for 3D playback and can be extended to include others.

#### 4.6.4.1 Quality Evaluation Tests

The users evaluated test sequences at a variety of qualities. The 10 second test sequences were presented in a random order on the 15-inch Sharp autostereoscopic 3D displays with a 5 second gray level image in between the test sequences. Figure 4.33 shows the presentation order used in the experiments. Each participant evaluated a total of 34 ten second 3D clips. The experiments used two different sequences encoded at varying qualities. To evaluate the impact of asymmetric coding, the test sequences were encoded such that quality of one view of the stereo pair is kept constant at a high quality while the quality of the other stereo view is varied from high to low quality. We used video coded at 42.5 dB as a high quality point and the lowest quality video was coded at 28 dB. The tests were evaluated with 16 participants with eight left-eye dominant and eight right-eye dominant. The equal number of left and right eye dominant participants is a coincidence and was not by design. The dominant eye test was conducted using the commonly used hole-in-the-card test. The data collected included handedness and eyedness.

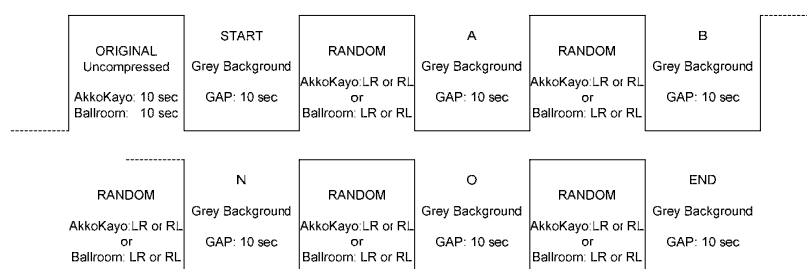


Figure 4.33 Timing of subjective 3D Image Quality of each random constructed video set

#### 4.6.5 Results and Discussion

The quality of the 3D video experienced primarily depends on the coding artifacts present in the individual views and the type of 3D display. The influence of the different types of artifacts present in the individual views is not well understood. The quality of a single 2D view alone is not an indication of the 3D quality. Developing objective quality metrics

for 3D quality is thus very difficult and subjective evaluation is the primary means of evaluating 3D video quality.

#### 4.X.5.1 3D Video Quality and Eye Dominance

While it has been known that human have a preference of one eye over the other, the significance of this preference is not well understood. Humans are mostly right handed (90%) and about 70% are right eyed, 20% left eyed, and 10% exhibit no eye preference [12]. The larger number (50%) of left-eye dominant participants in the 3D evaluation can perhaps be explained by the fact that all the participants are from the college of engineering. A recent study suggested that the eye dominance just indicates individual sighting preferences and has no function in binocular vision [13]. A more recent study, however, found that eye dominance improves the performance of visual search tasks by perhaps aiding visual perception in binocular vision [14]. Our results also suggest a role for eye dominance in binocular vision.

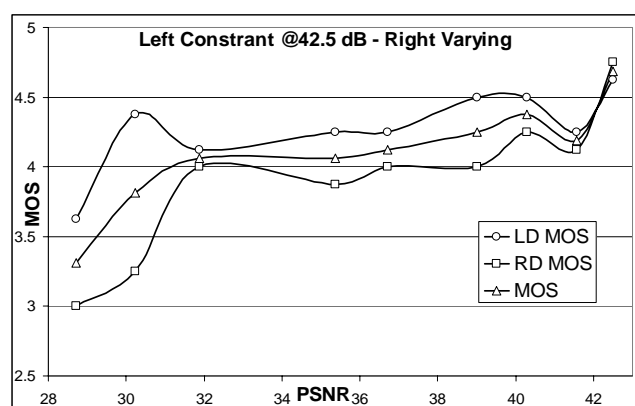


Figure 4.34 Mean opinion scores for asymmetric view coding with left eye view at a higher quality

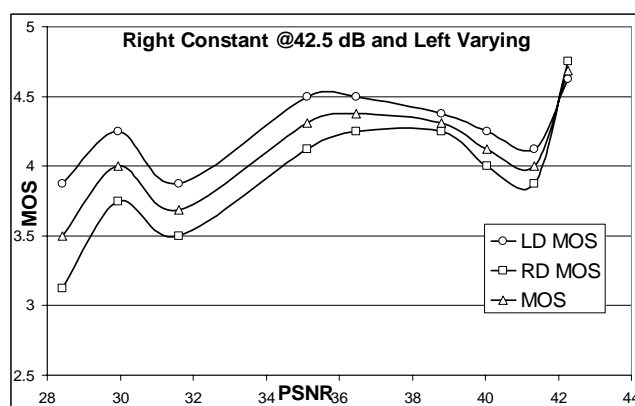


Figure 4.35 Mean opinion scores for asymmetric view coding with right eye view at a higher quality



Mean opinion scores were computed for the test sequences based on subjective evaluations. Figures 4.34 and 4.35 show the mean opinion scores (MOS) for the Akko and Kayo sequence with right eye view kept constant at 42.5 dB and the left eye view coded at lower qualities. A second set of sequences were also evaluated with left eye view encoded at 42.5 dB and right eye view quality varied from 42.5 dB to 28 dB. The figures show the MOS for all the users, the right-eye dominant users, and left-eye dominant users. The figures show that eye dominance does impact 3D perception. Right eye dominant users seem to be more sensitive to the asymmetric video quality. As the quality of the right (left) view increases, the difference between the left-eye and right-eye dominant users decreases.

The MOS is about one point higher for left-eye dominant users when one the views is encoded at a lower quality. The increased sensitivity of right-eye dominant users puts constraints on the lower bound of view quality in asymmetric view coding. Further study is necessary to understand why the right eye dominant users might be more sensitive to asymmetric video coding. The role of eye dominance has significant implications on the asymmetric view encoding of stereo views. The stereo views have to be encoded at a sufficiently high quality so that the right-eye dominant population does not experience poor 3D quality.

3D compression with H.264 view coding performs very well under asymmetric view coding. The binocular mixture in the human visual system suppresses this poor quality and gives the users a reasonably good 3D experience. The low quality left eye view in this case was encoded at a very low quality and is completely unacceptable by itself. As shown in the figure, the low quality left-eye view lost significant picture details due to quantization. The pattern on the background is lost and the facial features are completely blurred. However, when combined with a high quality right eye view, the 3D/depth perception is well preserved. The resulting 3D view has blocking artifacts on the background but contains all the background and foreground details that are lost in the left-eye view.

Binocular vision is not the only source of depth perception. The monocular views contain depth cues which are combined with the disparity information to give the depth perception. The asymmetric view coding principle can be further exploited by coding the low quality view such that the visual cues that contribute to depth perception are coded with a higher quality compared with the regions without any depth cues. Similarly, flat regions in a picture (regions without depth) can be compressed more than the regions with objects present. The presence of an edge is one simple metric that can be used to drive such adaptive compression in asymmetric view coding. The blocks with edges can be coded with higher quality compared to the edge-free blocks in the picture. The impact of these adaptive coding techniques on the eye dominance also needs to be studied.

### 4.6.6 Algorithms for Multi-View Video Coding

Figure 4.36 shows the general architecture of the multi-view video coding system based on the JSVM reference software used by the MPEG Multi-view standardization group.

This architecture supports N views of the same scene and encodes the views by exploiting the large amount of redundancies among the views. We use H.264 as the core compression engine with inter-view prediction to increase compression efficiency.

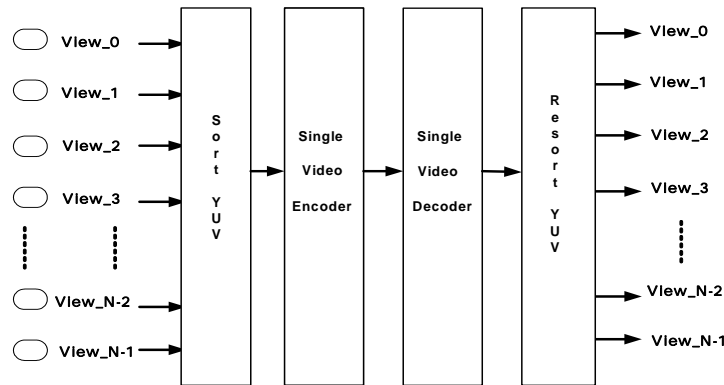


Figure 4.36 Multi-View video coding architecture

#### 4.6.6.1 A New Hypercube Prediction Algorithm

We propose a novel prediction algorithm for MVC that balances the compression efficiency and decoder complexity. The proposed algorithm, hypercube prediction algorithm (HPA), derives prediction dependencies based on a hypercube layout. We have evaluated this prediction algorithm using the JSMV 3.5 software, which is the reference implementation of the MPEG MVC standards effort.

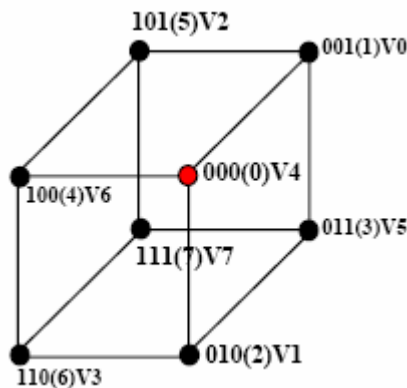


Figure 4.37 3rd Order Hyper-Cube Algorithm

The proposed design in Figure 4.37 illustrates a new way to implement multiple view cameras encoding and decoding. A prediction algorithm based on the hypercube structure was developed to improve the performance of MVC systems. The camera views are mapped to the nodes of a hypercube and the dependencies are derived based on the position of the camera. The dependencies are communicated to the receiver by providing a node-camera map. Figure 4.37 shows the nodes of a hypercube and the corresponding camera maps for a 1D array of 8 cameras.

This type of Hypercube structure allows a well structured dependency description. The binary node IDs, the equivalent n-bit binary number in parenthesis, and the view number  $V_i$  is shown in Figure 1. The reference views for a view  $V_i$  are derived as follows:  $V_i$  can use a view  $V_j$  for prediction if  $V_i$  and  $V_j$  are adjacent and  $V_i > V_j$ .

Table 4.7 Reference views for an eight camera array

View No. ( $V_i$ )	Ref.Views ( $V_j$ )	No. Views in dependency chain	Ref.Views ( $V_k$ )	No. Views in dependency chain
0 (000)	-	-	-	-
1 (001)	0	1	0	1
2 (010)	0	1	1	2
3 (011)	1,2	3	2	3
4 (100)	0	1	3	4
5 (101)	1,4	3	4	5
6 (110)	2,4	3	5	6
7 (111)	3,5	4	6	7
Total		13		21
Average		13/8=1.52		28/8=3.5

Figure 4.38(a) shows a basic prediction structure called Linear Prediction Algorithm (LPA). In LPA, a view forms prediction from a view immediately to its left. Figure 4.38(b) shows the prediction dependencies for an 8-camera MVC system using the Hypercube Prediction Algorithm (HPA). The LPA has simple structure where the interview prediction is based on the view to the left. This creates a long dependency chain that increases with the number of cameras. For example, with the LPA prediction, view 7 depends on all the other views. Table 4.7 shows the reference views, number of views in the dependency chain, and the average number of view dependencies.

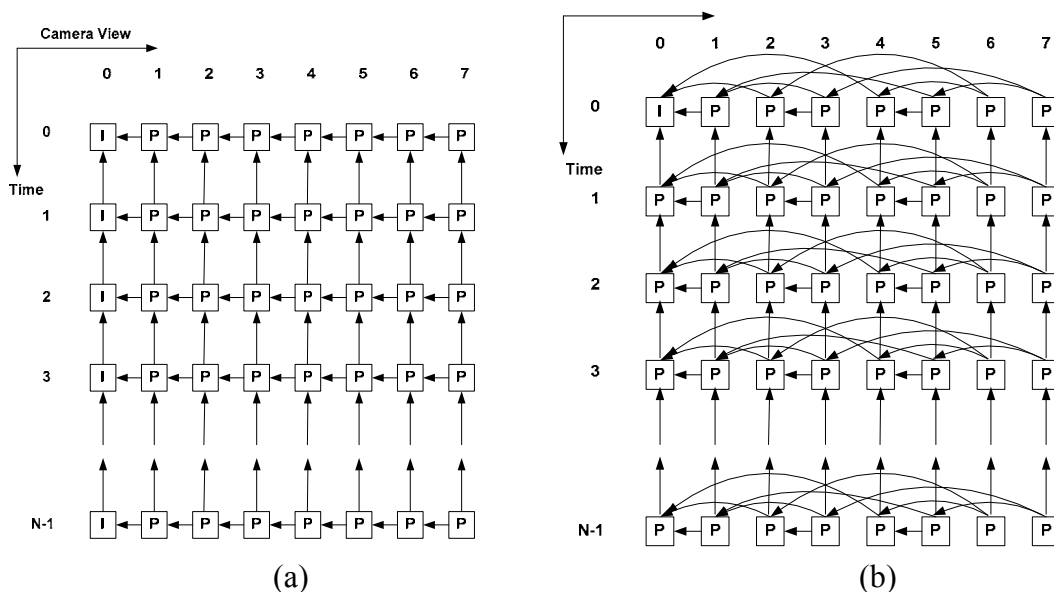


Figure 4.38 Ballroom Camera View-Time Grid for MVC (a) Prediction Algorithm (b) Hypercube Prediction Algorithm

The view dependencies for the HPA are derived based on the Hypercube node mapping as shown in Table 4.x.1. The hypercube structure determines the view dependencies and a camera-to-mapping is done such that the dependent view are strongly correlated. This means view 0 is not mapped to node 0 of the hypercube. Node 0 of the hypercube is a middle view that can have other views depend on it. Optimal view mapping requires taking the camera geometry into consideration. Table 4.7 also shows the average number of view dependencies (AVD) for LPA and HPA. The AVD metric gives the average number of views that must be decoded to play a view. The smaller AVD leads to lower complexity decoders but could affect the RD performance. As shown in Table 4.x.1, the HPA coded content requires half the number of view on average compared to LPA. The RD performance has to be evaluated experimentally to understand the impact of lower AVD on the RD performance.

#### 4.6.6.2 Experimental Setup

Experiments were conducted to evaluate the RD performance of the HPA and LPA. The experiments were done using JSVM 3.5 modified for multiview coding. The number of reference frames used in the HPA changes with the view and the encoder was modified to change the number of active reference frames depending on the view. The number of reference frames is always 2 for the LPA. The decoded picture buffer size is set to 8 as the prediction uses a frame that is at most 8 frames away. Only I and P frames were used. The reference pictures were set using the reference picture reordering list (RPLR) commands and the Format String used by JSVM. The camera to node mapping for this 8 camera arrangement is as shown in Figure 2. Here the mapping of Hypercube nodes to the camera views was chosen such that we obtain maximum prediction improvement and minimize the temporal distance.

For the experiments, we used the Ballroom multi-view video sequences at different QP values. The video sources are 8.33 seconds long, 640x480 resolution, 30 FPS, 250 frames in length, and available in YUV 4:2:0 format. Initially, we started with eight camera views to study the algorithm performance. Only the first frame of view 0 is coded as I frame and the rest are coded as P frames. The first row in Figures 3 and 4 represents the eight camera views at time instant 0, which implies the first picture of the view to be encoded. The pictures in the same view have temporal dependencies and spatial redundancies are exploited across views.

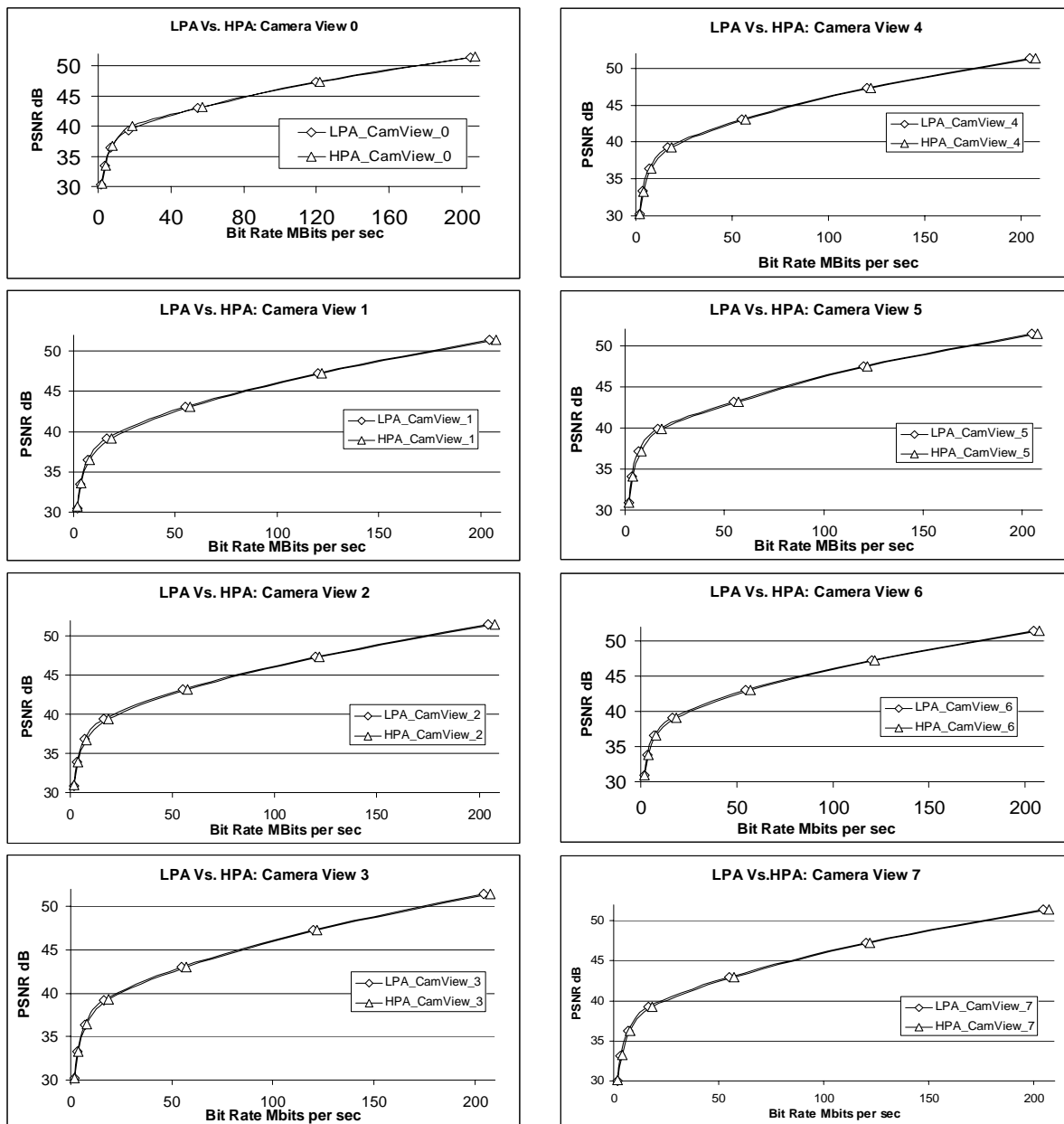


Figure 4.39 RD performance of the HPA and LPA for the 8 views

#### 4.6.6.3 Results and Discussion

We present some experimental results that illustrate the benefits of the HPA algorithm. The LPA with the prediction structure shown in Figure 4.39 and HPA with prediction structure shown in Figure 4.39 were both evaluated. Note that because of node-to-camera mapping in HPA, the view 0 shown in Figure 4.38(b) actually maps to view 4. The RD plots comparing the HPA and LPA for each view are shown in Figure 4.39. The bitrate shown is the combined bitrate of all the views.

Extensive experimental results show that the RD performance of the HPA and LPA is very close. However, the prediction structure used in HPA reduces the average number of views to be decoded to half that of LPA. A general observation for all camera views is that the same temporal prediction for both LPA and HPA. Consequently, the main parameter that determines the RD-performance is the spatial prediction. The RD performance of the HPA can be improved by using alternative camera-to-node mappings made possible because of the flexibility of the HPA.

#### **References for Section 4.6**

- [1] B.L. Tseng and D. Anastassiou, "Multi-Viewpoint Video Coding with MPEG-2 Compatibility," IEEE Trans. Circuits and Systems for Video Tech., Vol. 6, No. 4, Aug. 1996, pp. 414-419.
- [2] A. Puri, R. V. Kollarits and B. G. Haskell, "Stereoscopic Video Compression Using Temporal Scalability," Proc. SPIE Visual Communications and Image Processing'95, Taiwan, May 1995.
- [3] Lew B. Stelmach, W. James Tam, "Stereoscopic image coding: Effect of disparate image-quality in left- and right-eye views", Signal Processing: Image Communication, Vol. 14, pp.111-117, 1998.
- [4] N.A. Dodgson, "Autostereoscopic 3D Displays," Computer, Volume 38, Issue 8, Aug. 2005 pp. 31 – 36.
- [5] L.M.J. Meesters, W.A.Jsselsteijn, and P.J.H. Seuntings, "A Survey for Perceptual Evaluations and Requirements of Three-Dimensional TV," IEEE Trans. Circuits Syst.Video Technol., Vol. 14, No. 3, Publisher, Location, pp. 381-391, March 2004.
- [6] H. Kalva and B. Furht, "Hypercube based inter-view prediction for multi-view video coding," Proceedings of the 2nd Workshop on Immersive Communication and Broadcast Systems (ICOB), October 2005.
- [7] O. Schreer, P. Kauff, and T. Sikora, eds., "3D Video Communications" Wiley 2005.
- [8] Daniel V. Meegan, Lew B. Stelmach, and W. James Tam, "Unequal Weighting of Monocular Inputs in Binocular Combination: Implications for the Compression of Stereoscopic Imagery", Journal of Experimental Psychology: Applied, Vol. 7(2) 143-153, Jun 2001.
- [9] C. Fehn, "A 3D-TV approach using depth-image-based rendering (DIBR)", Proc. of VIIP 03, Benalmadena, Spain, Sept. 2003.
- [10] AviSynth, <http://www.avisynth.org>

- [11] ISO/IEC JTC1/SC29/WG11, “Survey of Algorithms used for Multi-view Video Coding (MVC),” MPEG Document MPEG2005/N6909, January 2005.
- [12] D.C. Bourassa, I.C. McManus, and M.P. Bryden, “Handedness and eye-dominance: A meta-analysis of their relationship,” *Laterality*, Vol 1, No. 1, 1996, pp. 5–34.
- [13] A.P. Mapp, H. Ono, and R. Barbeito, “What does the dominant eye dominate? A brief and somewhat contentious review,” *Perception & Psychophysics*, Vol. 65, No. 2, 2003, pp. 310–317.
- [14] E. Shneor, and S. Hochstein, “Eye dominance effects in feature search,” *Journal of Vision*, 5(8), 699a, <http://journalofvision.org/5/8/699/>, doi:10.1167/5.8.699.

## **4.7 Object Segmentation Using Depth Information**

This report summarizes our findings on the feasibility of enhancing object segmentation using depth information. It builds on a foundation of work dealing with the segmentation of objects in traditional, two-dimensional images and improves on these methods by incorporating stereo disparity. Complicated cases in two-dimensional image segmentation, such as occluding objects that are similarly textured, colored, and shaded reduce the accuracy of these solutions. However, a disparity map may make the segmentation of these occluding objects much easier as they cannot exist at the same depth.

In this report we present a new model for object segmentation using depth information. It extends previous work on a saliency-based region of interest (ROI) extraction method. Our new method demonstrates consistently improved results. It was designed with high-resolution imagery, particularly that available from the HDMAX camera, in mind. Our methods will improve as available image resolution increases.

### **4.7.1 Introduction**

Object segmentation in videos has been studied widely in literature. The availability of multi-view or stereo-view video sequences makes it possible to estimate the depth of objects in a scene. Using the depth information will improve the accuracy of the object segmentation algorithms. In this project we will develop algorithms, tools, and software for depth-based object segmentation in video sequences.

The work presented in this report focuses on evaluating the feasibility of extending this work using the additional information provided by a second, stereo view. Specifically, we improve previous work by using a generated disparity map.

The main objectives of this project are

- Investigation of the techniques for 3D object segmentation
- Implementation of selected algorithms for 3D object segmentation

The research question that has guided this work has been: “How can the latest research and best practices in computer vision and the related field of cognitive science be used to improve object segmentation when depth information is?” More specifically,

- Which aspects of image segmentation are relevant for coastline security?
- Which computer vision techniques are most suitable for the proposed tasks?
- How can knowledge from the domain of cognitive science improve object segmentation?
- What type of performance improvement can be achieved by implementing the proposed model?
- Which scenarios would benefit most from the proposed solution?

### 4.7.2 Background

The proposed method combines a saliency-based ROI extraction method with depth map information. This Section presents background information on relevant topics.

#### 4.7.2.1 Vision science

Much of the visual information our eyes sense is discarded. Instead, our brain prioritizes what points in a scene we focus our attention on. The result is a series of fixations and saccades known as scanpaths. There are two ways attention manifests itself; bottom-up and top-down. The former is rapid, involuntary, and in reaction to the stimulus which is presented [27]. Only later does top-down attention take place. It is motivated by our past knowledge and memories [27]. Both play a role in how our attention is ultimately guided, but to what extent remains unclear. However, since top-down attention is a complex process, the computational modeling of bottom-up processes of visual attention has been most successful to date. Our interest in vision science is set out in this section.

Our previous work [13] demonstrated a method of extracting regions of interest based on their saliency. It integrated the Itti- Koch bottom-up computational model of visual attention [12] and that from Stentiford [26] through a series of morphological operations. The model produces one or more extracted regions of interest.

The field of vision science is diverse and broad. For this reason, this work will consider relatively focused topics from this area. Vision science is the study of how humans (and other life) see and interpret the light that lands on the sensor known as the retina. [18] describes the physiology of the human visual system (HVS). Two key topics are relevant to this study:

- Attention: how does the HVS select what it does for processing?
- Perception: how does the HVS interpret what it sees?

It is not possible for the human visual system to consider an entire image at once. Rather, we rapidly select several points-of-attention to direct our vision at when presented with a new scene. As a result, most of the light that radiates and falls upon our retinas is ultimately ignored. Like with other senses, our brain acts as a filter that greatly reduces



the amount of stimuli we perceive at any one time. We can focus in on a voice in a crowd or ignore the sensation our clothes make against our skin. Similarly, unless we specifically pay attention to certain elements in a visual scene only those areas of a scene that are salient or relative to the active visual search task will be attended to. In order to accomplish this, our eyes make a rapid series of movements known as scanpaths [16]. This ability to prioritize our attention is not only a matter of efficiency, but critical to survival.

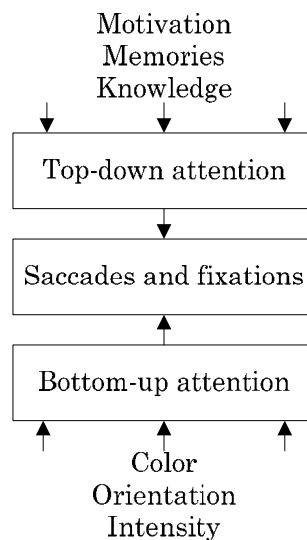


Figure 4.39 Examples of factors that influence attention

Attention can either be bottom-up or top-down (see Figure 4.39). While each is well-defined, there remains a gray area in that there are cases where we are not sure if top-down or bottom-up factors are responsible for attention, nor do we know with certainty how the two interact. Bottom-up attention is rapid and involuntary – it is an instinct. In general, bottom-up processing is motivated by the stimulus presented [27]. Our immediate reaction to a fast movement, bright color, or shiny surface is performed subconsciously and automatically without any consideration. Features of a scene that influence where our bottom-up visual attention is directed are the first to be considered by the brain and include color, movement, and orientation, among others [12]. For example, we impulsively shift our attention to a flashing light, regardless of our current task. This salient point, if it furthers our objective, may have more consideration devoted to it – a top-down process. Top-down attention is influenced by knowledge – what we have learned and can recall. Top-down processing is initiated by memories and past experience [27]. Looking for a specific letter on a keyboard or the face of a friend in a crowd are tasks that rely on learned, top-down knowledge.

Ultimately, both bottom-up and top-down factors contribute to how we choose to focus our attention. However, the extent of their interaction is unclear. Unlike attention that is influenced by top-down knowledge, bottom-up attention is a consistent and purely biological process. In the absence of top-down knowledge, a bright red stop sign will

instinctively appear to be more salient than a flat, gray road. Computational modeling of visual attention has made the most progress interpreting bottom-up factors that influence attention whereas the integration of top-down knowledge into these models remain an open problem. Not only do bottom-up components of a scene influence our attention before top-down knowledge does [4], but bottom-up attention can be overridden by top-down goals.



Figure 4.40 An image, its saliency map (Itti-Koch), and its attention map (Stentiford)

In 1998 Itti, Koch, and Niebur published their model of saliency-based visual attention [12]. Since then, the model has been used in many diverse applications, from directing robots to analyzing the quality of magazine layouts. This work applies their model to content-based image retrieval.

The Itti-Koch model of visual attention considers the task of attentional selection from a purely bottom-up perspective, although recent efforts have been made to incorporate top-down impulses. The model generates a map of the most salient points in an image (“the saliency map”). Color, intensity, orientation, motion, and other features may be included in the saliency computation. Figure 4.40 (center) shows an example of the Itti-Koch saliency map which considered color, intensity, and orientation.

The saliency map produced by the model can be used in several ways. It has been applied to identify regions-of-interest by using the most salient points as cues [13]. Rutishauser et al. [23] employ the Itti-Koch model to extract regions examining the area around the most salient patch of an image and then using region-growing techniques. Key points extracted from the detected object are used for object recognition. Repeating this process after the inhibition of return has taken place enables the recognition of multiple objects in a single image. Inhibition of return is the suppression of recently viewed areas, regardless of their saliency. The model has also been used in the context of object recognition [28]. Navalpakkam and Itti have begun to extend the Itti-Koch model to incorporate top-down knowledge by considering the features of a target object [14]. These features are used to bias the saliency map. For instance, if one wants to find a red object in a scene, the saliency map will be biased to consider red more than other features.

The ability of the Itti-Koch saliency model to actually predict human attention and gaze behavior has been analyzed elsewhere ([7], [19], [20], [21]) and is not free of criticism. It is not difficult to find cases where the Itti-Koch model does not produce results that are

consistent with actual fixations. The work of Henderson et al. documents one such instance where the saliency map (and other computational models of visual attention) does not share much congruence with the eye saccades of humans [10]. However, this work adds the constraint that the visual task being measured is active search, not free viewing. The Itti-Koch model was not initially designed to include the top-down component that active search and similar tasks require.

The model of visual attention proposed by Stentiford [26] (referred to as the Stentiford model) is also biologically inspired. It functions by suppressing areas of the image with patterns that are repeated elsewhere. As a result flat surfaces and textures are given low scores while unique objects are given prominence. Regions are marked as high interest if they possess features not frequently present elsewhere in the image. The result is a visual attention map that is similar in function to the saliency map generated by Itti-Koch, but quite different in appearance and applicability.

The visual attention map generated by Stentiford tends to identify larger and smoother salient regions of an image, as opposed to the more focused peaks in Itti-Koch's saliency map. Stentiford's model is more suited to segmentation rather than detection of salient regions, as described in [13]. Unfortunately, the tendency of the Stentiford model to mark large regions as being salient can lead to poor results. Itti's model is better in this regard. Refer to [1] for a more detailed description of the Stentiford model.

In 1998 Rybak et al. proposed a computational model of visual perception and recognition that is led by attention [24]. Their work generates scanpaths for a good representation of an image. It is demonstrated that this scanpath can be used to recognize images invariantly.

In [6] Draper et al. show that even a simple implementation of visual attention (in their case, detecting corners) can yield useful results. They model the expert object recognition pathway which is the part of the brain that recognizes specific object. Attention is used to feed data points to this pathway. Ultimately, this results in hierarchical categories.

Once the human visual system selects what merits further inspection, the human visual system's perceptual abilities must interpret the stimuli. Perception is the processing of these senses [27]. Perception occurs in a variety of specialized areas in the brain. For example, the identification of human faces causes activity in the fusiform gyrus [15]. While the exact stages of recognizing faces remain unknown, we do have a general ideal of the basic stages that contribute to the process. Farah et al. review the literature to determine that face recognition is different from other kinds of object recognition [8]. From this work it can also be inferred that other recognition tasks, such as reading, rely on yet other specialized parts of the brain. Taken to the extreme, words are not perceived as faces – why is this so? Our brain has evolved and is trained through experience to (relatively) immediately distinguish such cases.

A key challenge of computer vision (and vision in general) is that a single stimulus (pattern of light) may have multiple interpretations. In humans and in most computer

systems the stimulus consists of one or more two-dimensional projections of a three-dimensional world. Naturally, there is considerable information that is lost in this translation.

For the human rapid recognition and interpretation of a scene depends on context. Indeed, determining the context, also known as the gist of a scene can occur even without attention [17]. Still, in most visual processing tasks attention is needed prerequisite for perceptual processing. Once context is determined our memories and acquired rules (knowledge) lead to expectations of the visual environment [22]. These expectations can be extremely powerful in eliminating potential interpretations of a scene and can even be difficult to overcome despite overwhelming evidence indicating a different interpretation is valid. There are two notable ways this can occur.

- Priming: [18] demonstrates that humans are more successful in identifying an object if it is preceded by relevant information. In this particular example, a mailbox and loaf of bread both are drawn very similarly (of not identically). When primed with an image of a kitchen a loaf of bread is identified. When primed with an outdoor scene the same figure is interpreted as a mailbox.
- Expected spatial location: Biederman's hydrant [2] is a classic experiment in which he demonstrates the difficulty people have in identifying objects if they do not occur at the expected position. In this case a fire hydrant is drawn floating in the air rather than fixed to the ground. He shows that participants take notably longer to identify the oddly located hydrant.

Optical illusions can occur if multiple interpretations share the same likelihood of occurring and thus cannot be resolved. An example of this is the famous Rubin vase in which our interpretation oscillates between seeing a vase and seeing two faces, never settling. The reader is referred to [18] for a discussion of this and other illusions and the Gestalt principles that are responsible. To date, computer systems do not have this "flaw" in human vision. Indeed, it has even been proposed that visual illusions that would fool a human but be imperceptible to a computer vision system be used as a type of Turing test [25] for applications such as authentication and steganography [3].

#### 4.7.2.1 Two-dimensional region-of-interest extraction

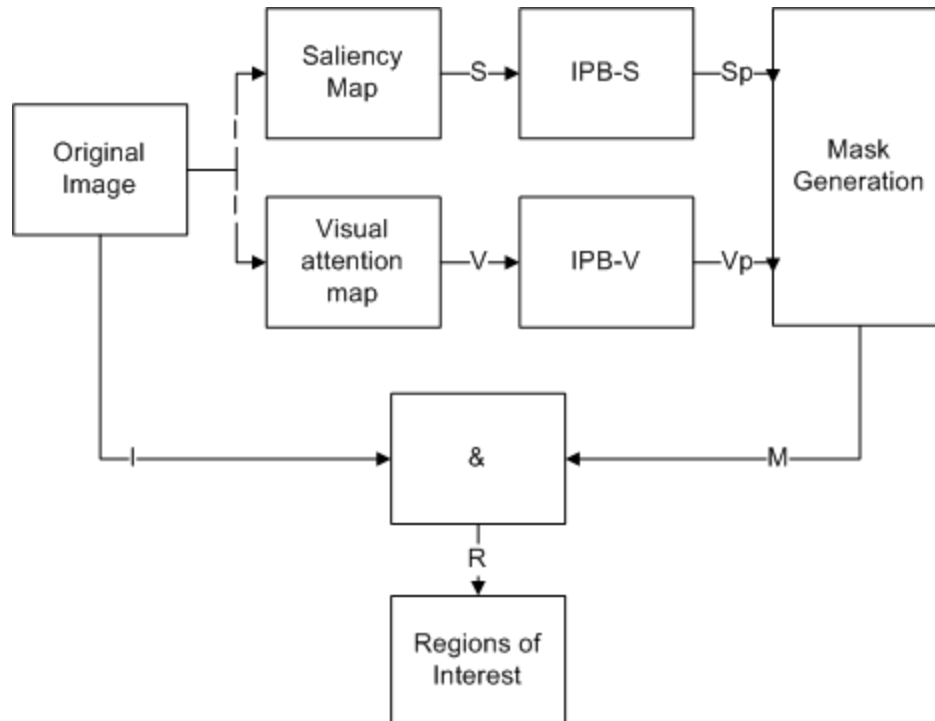


Figure 4.41 General block diagram of the 2D ROI extraction method

Figure 4.41 shows an overview of the 2D ROI extraction method. The saliency map (Itti-Koch) (S) and visual attention map (Stentiford) (V) are generated from the original image. Post-processing is performed independently on each in order to remove stray points and prune potential regions. Then, the remaining points in the processed saliency map are used to target regions of interest that remain on the visual attention map. The result is a mask (M) that can be used to extract the regions of interest (R) from the original image. This process is detailed in [13].

We incorporate a model of visual attention to compute the salient regions of an image. Regions of interest are extracted depending on their saliency. Our first cue is the salient peaks in the Itti-Koch saliency map. If these peaks overlap with salient regions in Stentiford's model, we proceed to extract a region of interest around that point. Images are then clustered together based on the features extracted from these regions. The result is a group of images based not on their global characteristics (such as a blue sky), but rather on their salient regions. When a user is viewing scenes or images the salient regions are those that stand out more quickly.

The model has four key aspects:

- Biologically plausible: combines the Itti and Koch's and Stentiford's biologically inspired models of visual attention
- Unsupervised: human interaction is not required

- Bottom-up: top-down knowledge, to date, has not been adequately incorporated into models of visual attention
- Modular: individual components of the model can be replaced, potentially improving overall system performance

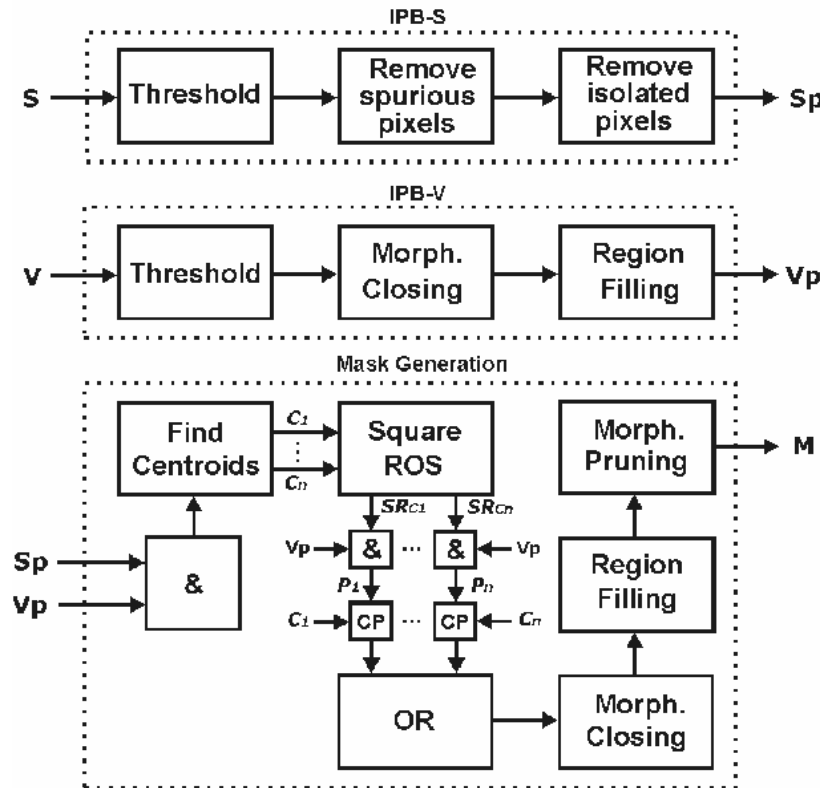


Figure 4.42 Region of interest extraction: detailed block diagram

Figure 4.42 details the inner workings of the IPB-S, IPB-V and mask generation blocks. The IPB-S block performs the following operations:

- Thresholding: converts a grayscale image  $f(x, y)$  into a black-and-white (binary) equivalent. This is accomplished by using the “`im2bw()`” function in MATLAB.
- Remove spurious pixels: removes undesired pixels from the resulting binarized image. This is implemented using a binary morphological operator available in the “`bwmorph()`” function (with the spur parameter) in MATLAB.
- Remove isolated pixels: removes any remaining white pixels surrounded by eight black neighbors. This is implemented using a binary morphological operator available in the “`bwmorph()`” function (with the clean parameter) in MATLAB.

The IPB-V block performs thresholding (as explained above) followed by the two operations below.

- Morphological closing: fills small gaps within the white regions. This is accomplished by using the “`imclose()`” function in MATLAB.

- Region filling: flood-fills enclosed black regions of any size with white pixels, starting from specified points. This is implemented using a binary morphological operator available in the “imfill()” function (with the holes parameter) in MATLAB.

The mask generation block performs (self-explanatory) logical AND and OR operations, morphological closing, and region-filling (as described above) plus the following steps.

- Find centroids: shrinks each connected region until only a pixel is left. This is accomplished by using the “bwmorph()” function (with the shrink parameter) in MATLAB.
- Square relative object size (ROS): draws squares of fixed size (limited to 5% of the total image size) around each centroid.
- CP: combines each centroid image (C) with a partial (P) image in order to decide which ROIs to keep and which to discard.
- Morphological pruning: performs a morphological opening and keeps only the largest remaining connected component, thereby eliminating smaller (undesired) branches.

There are certain cases where the aforementioned method does not work. When objects are occluded or overlapping they may appear as a single region when inspecting a single 2D projection of the view. Only with a separate view can enough information of the original 3D scene be reconstructed to determine the relative depth of the occluding objects. Conversely, relying only on depth information is also not enough to properly determine a region of interest. A bright poster on a flat wall, for example, would be ignored if only depth information were used, as it rests on the same plane as the wall. As a result, we propose a combination of both methods, mitigating the weaknesses of each.

#### 4.7.2.1 Stereo vision and the disparity map

Given a pair of stereo images, the correspondence problem refers to finding the match sequence for each left and right image scanline. The match refers to an ordered pair  $(x; y)$ , where  $x$  and  $y$  are the positions in same scanlines of left and right stereo pair, respectively, such that the pixel values corresponding to these positions represent images of the same scene point. Here, it is assumed that the stereo images are properly aligned so that the scanlines are the epipolar lines. Unmatched pixels are labeled as occluded, and adjacent occluded pixels bounded by non-occluded pixels are called an occlusion.

The disparity of a pixel position  $x$  in the left scanline that matches the pixel  $y$  in the right scanline is defined as the difference  $x - y$ , while the disparities of the pixels in an occlusion are assigned the farther of the two bounding regions. Approaches to the stereo correspondence problem construct the so called disparity map, which is also often called the depth map or the depth estimation since it describes the discrete estimation of third spatial dimension.

In [31], the authors proposed fast and effective algorithm for depth estimation from stereo images. Unlike other similar approaches, such as [5] [9] [32], the approach of Birchfield and Tomasi achieves optimal performance mainly by avoiding sub-pixel resolution with a

measure that is insensitive to image sampling. The depth estimation phase of our method relies on this computational approach. Details of Birchfield-Tomasi algorithm can be found in [31], while [30] contains a detailed description of the proposed measure.

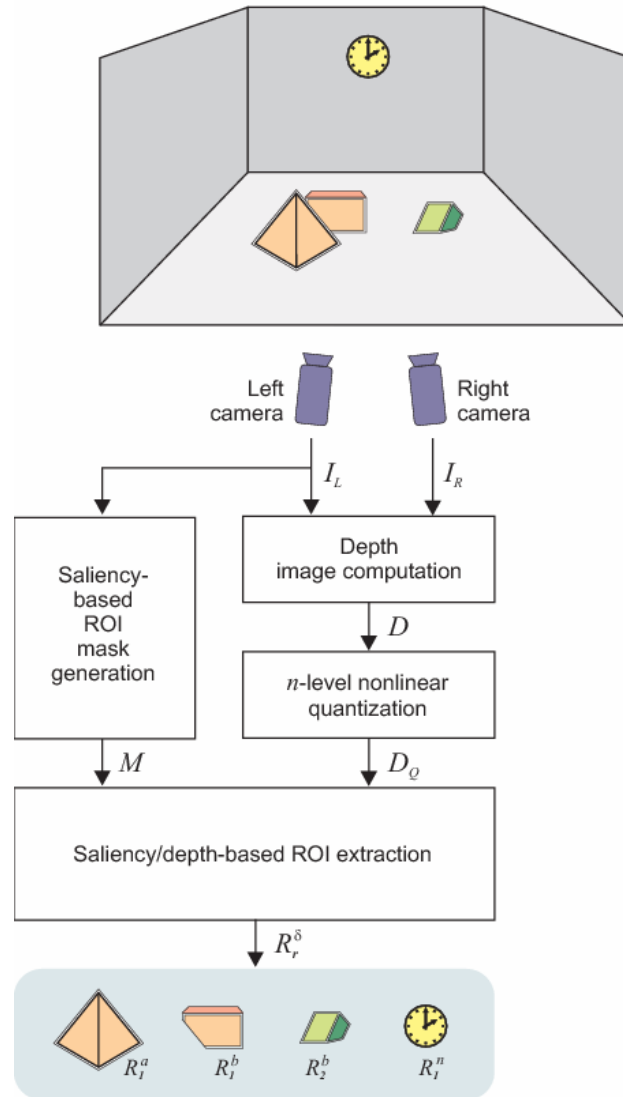


Figure 4.43 General block diagram of the 3D ROI extraction

According to Figure 4.43, the scene is first acquired by two properly-positioned and adjusted cameras, so that the scanlines are the epipolar lines. The left and right stereo images,  $I_L$  and  $I_R$  are processed by Birchfield-Tomasi disparity estimation algorithm. The output disparity map  $D$  is then nonlinearly quantized within  $n$  levels, resulting in output image  $D_Q$ . The left channel image,  $I_L$ , is also processed by the existing 2D saliency-based ROI segmentation algorithm that produces a binary mask  $M$  corresponding to the salient regions of the image (Figure 3.39). In the last stage of the algorithm,  $M$  and  $D_Q$  are submitted to the saliency/depth-based ROI extraction block, which combines both images in order to segment the ROIs and label them according to



their respective depths in the real scene.  $\delta$  is the quantized depth, with  $\delta$  belonging to  $\{a, \dots, n\}$  and  $r$  is an ROI at a depth  $\delta$ .

In the example shown in Figure 4.43, the objects (ROIs) belong to foreground, middle, or background. In the output at the bottom of the figure the pyramid within the foreground plane is labeled with  $R_{a1}$ , the partially occluded parallelepiped and the green solid, at the same middle plane, are labeled with  $R_{b1}$  and  $R_{b2}$ . Finally, the clock in the background is labeled with  $R_{n1}$ .

Under normal conditions depth images are relatively efficient in discriminating objects at the frontal planes of the scene but they generally do not have sufficient resolution to capture flat objects in the background or even common objects on a distant plane. On the other hand, a saliency-based ROI identification algorithm can capture such objects, but they do not account for relative object depth within the scene. The objective is to combine the information provided by both salient regions and depth cues to improve ROI extraction.

In Figure 4.43, a purely saliency-driven ROI extraction algorithm tends to identify both light-orange objects as a single region. However, using depth information, it is possible to divide this region, discriminating the two objects. Another benefit of this approach is the possibility of extracting objects such as the watch in the background of Figure 4.43. While algorithms for depth estimation are not able to discriminate the watch plane from the wall plane (their depth is too similar), a saliency-driven ROI extraction can segment that object. Using only depth images the watch would not be captured.

### 4.7.3 The Proposed Model

The following is a description of the system components from Figure 4.43.

**Depth images:** The disparity maps generated by the Birchfield-Tomasi method are represented as 256-level grayscale images. Darker (lower) values indicate further distances, and vice versa. In particular, purely black values denote the background plane.

**Nonlinear quantization:** An  $n$ -level ( $L_1, \dots, L_n$ ) quantization is obtained and applied to the disparity map. Level  $L_1$  identifies the depth closest to the cameras and level  $L_n$  denotes the depth farthest depth from camera (the background).

$$D_Q(x,y) = \begin{cases} L_n & \text{if } D(x,y) = [0 \ T_1), \\ L_{n-1} & \text{if } D(x,y) = [T_1 \ T_2), \\ \vdots & \\ L_1 & \text{if } D(x,y) = [T_{n-1} \ 255]. \end{cases}$$

where  $T_i$  are the selected threshold values.

**Saliency-based ROI mask:** Salient regions of interest are extracted from the left image using the method described in [33]. This method was modified in the original saliency-

driven ROI extraction algorithm to refine some of the thresholds used to determine relative object size.

**ROI extraction:** The ROI extraction stage combines images M and DQ. Its goal is to segment and label the ROIs according to their depths in the real scene. First, an AND operation between grayscale image DQ and mask M is performed, originating a grayscale D image. This image is then used to perform depth decomposition.

$$\mathcal{D}^\delta = \begin{cases} 1 & \text{if } \mathcal{D}(x,y) = L_\delta, \\ 0 & \text{otherwise} \end{cases}$$

After that, ROIs can be effectively extracted. First, decomposed depth image D1 is submitted to a set of morphological operations, denoted by  $m(\cdot)$ .

$$R^1 = m(\mathcal{D}^1)$$

R1 is a binary image where the white regions correspond to ROIs into depth 1, that is, those that are closest to the camera. Function  $m(\cdot)$  performs the following sequence:

1. Closing: fills small gaps within the white pixels regions. Implemented using the `imclose()` function in MATLAB.
2. Region filling: flood-fills enclosed black pixels regions. Accomplished using the `imfill()` MATLAB function.
3. Pruning: performs a morphological opening and keeps only the largest remaining connected component, thereby eliminating smaller (undesired) branches.
4. Small blobs elimination: removes unconnected regions with area smaller than affixed number of pixels.

The remaining  $R^\delta$  for each decomposed depth are sequentially computed, from  $\delta=2$  to  $\delta=n$ :

$$R^\delta = m\left(\mathcal{D}^\delta \cap \left[\bigcup_{k=1}^{\delta-1} R^k\right]^c\right)$$

where  $[\cdot]^c$  means the complement operation. Note that the computation of a deeper  $R^\delta$  takes into account the depths before it. This operation gives preference to closer regions of interest over the further ones. Each image  $R^\delta$  can have a set of ROIs, denoted by:

$$\{R^\delta\} = \{R_1^\delta, \dots, R_r^\delta\}$$

where  $r$  is the number of ROIs in the depth  $\delta$ , with  $r \geq 0$ .

#### 4.7.4 Ongoing Work

We are currently working on extending the results of this work to the goals set for the third year of this grant, which focuses on building a complete visual surveillance solution

for coastline security needs, with emphasis on detecting suspicious behavior. In the proposed solution, whose framework is based on Figure 4.44 below, after objects of interest have been segmented – using a combination of color-, motion-, and depth-based information – they are classified and tracked across multiple frames. If the object being tracked is a human, it is appropriately classified as such and its actions and behaviors are analyzed by another layer of human recognition algorithms. It is our goal that – at the end – it will be possible to answer questions such as:

- Are there people present in the video?
- Where are they coming from and where do they end?
- Are people moving quickly or slowly?
- What types of vehicles are in the video?
- How long did the vehicle stop before resuming the journey?

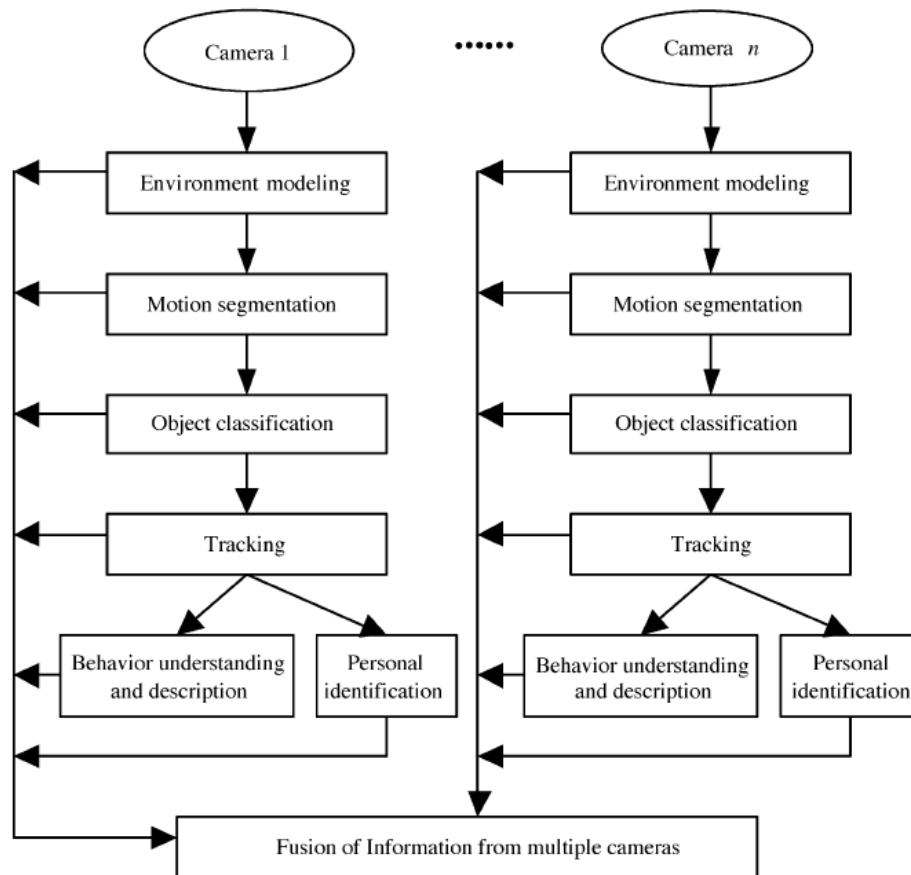


Figure 4.44 General framework of a multi-camera video surveillance system, from [29]

Reliable segmentation is an essential prerequisite of all subsequent steps, which makes the work described in this report relevant to the upcoming developments within the scope of this grant.

The following figures show an example of the work reported here in the context of upcoming (year 3) efforts. We are interested in tracking a person walking across the field of view (FOV) of the stereo-mounted cameras (Figure 4.45 and Figure 4.46). After having detected the presence of a moving foreground object of interest (in this case a person) within the FOV, our solution uses a combination of depth (Figure 4.47), motion (Figure 4.48), and edge-based (Figure 4.49) information to segment the object out of its surroundings, draw a bounding box around it and follow it along a number of frames (Figure 4.50).



Figure 4.45 Left (a) and right (b) views of a person shortly after appearing in the FOV



Figure 4.46 Left (a) and right (b) views of a person approaching the end of their appearance within the FOV

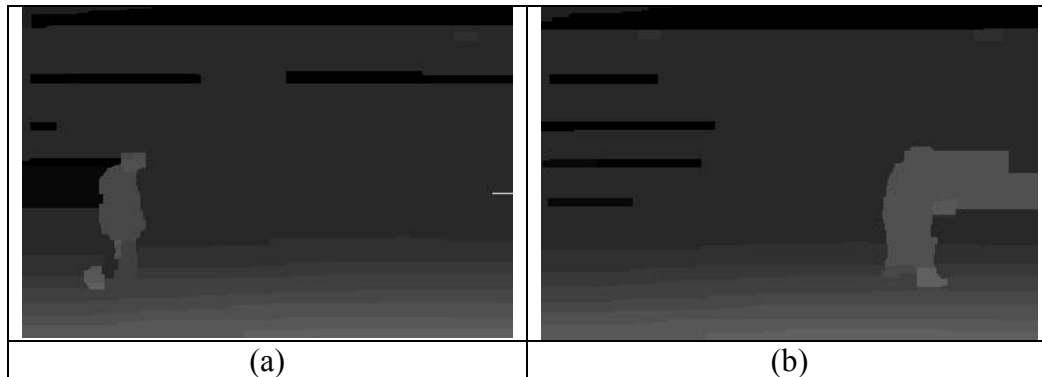


Figure 4.47 Depth-based information at the beginning (a) and end (b) of the walk. The object of interest appears brighter (meaning closer to the camera) than the background

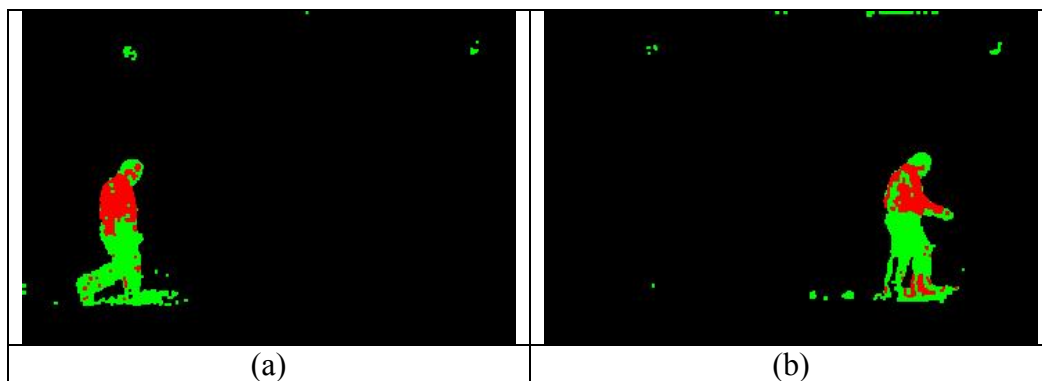


Figure 4.48 Motion-based information at the beginning (a) and end (b) of the walk; the object of interest appears pseudo-colored against a black background

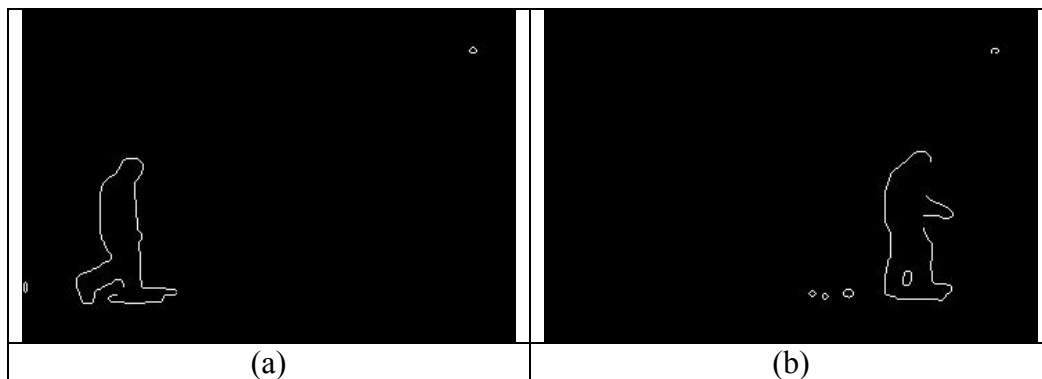


Figure 4.49 Edge-based results at the beginning (a) and end (b) of the walk; the object of interest appears pseudo-colored against a black background

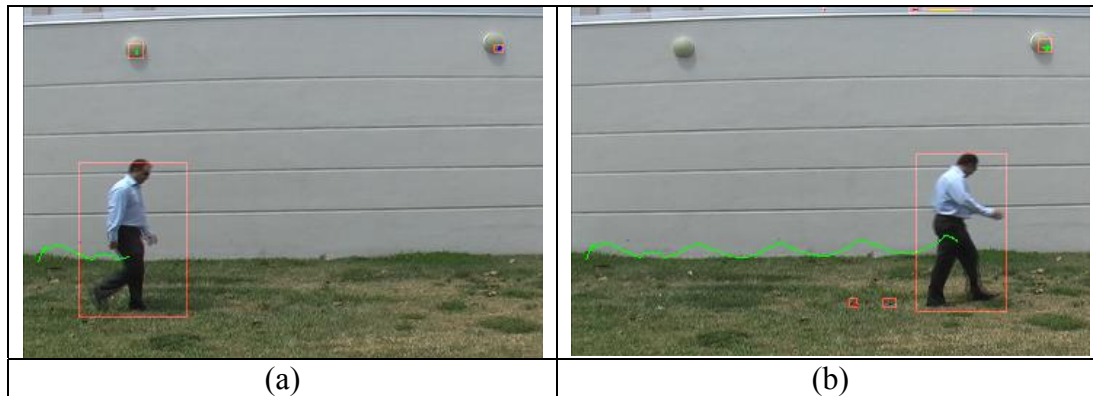


Figure 4.50 Tracking results at the beginning (a) and end (b) of the walk; the object of interest appears enclosed by a (red) bounding box and the trace of its trajectory is painted (in green) over the frame

#### 4.7.5 Conclusion

Object and region segmentation from 2D data is not always a straightforward task. In particular, it can be impossible to segment occluded object because of the depth information that is lost. In this work we extended a previously proposed method for 2D region of interest extraction with depth information. A disparity map was generated from two views using the method proposed by Birchfield-Tomasi [31]. Using this depth information we were able to differentiate occluding regions of interest. Our experiments demonstrate the promise of this approach but stress the need for nonlinear quantization thresholds of the disparity map for successful results. We are continuing work on this approach by creating a method of automatically determining these quantization thresholds and extending it to a variety of applications. We are currently obtaining quantitative results to further validate our method.

#### References for Section 4.7

- [1] Bamidele, A.; Stentiford, F. & Morphet, J. An attention-based approach to content-based image retrieval British Telecommunications Advanced Research Technology Journal on Intelligent Spaces (Pervasive Computing), 2004, 22(3)
- [2] Biederman, I. Perceiving real-world scenes Science, 1972, 177, 77-80
- [3] Brelstaff, G. and Chessa, F. 2005. Practical application of visual illusions: errare humanum est. In Proceedings of the 2nd Symposium on Applied Perception in Graphics and Visualization (A Coruña, Spain, August 26 - 28, 2005). APGV '05, vol. 95. ACM Press, New York, NY, 161-161. DOI= <http://doi.acm.org/10.1145/1080402.1080441>
- [4] Connor, C.; Egeth, H. & Yantis, S. Visual Attention: Bottom-Up Versus Top-Down Current Biology, 2004, 14, 850-852

- [5] Cox, I., Hingorani, S., Rao, S., and Maggs, B. (1996). A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567.
- [6] Draper, B.; Baek, K. & Boody, J. Implementing the expert object recognition pathway International Conference on Vision Systems, Graz, Austria, 2003
- [7] Einhauser, W. & Konig, P. Does Luminance-contrast Contribute to a Saliency Map for Overt Visual Attention? *European Journal of Neuroscience*, 2003, 17, 1089-1097
- [8] Farah, M.; Wilson, K.; Drain, M; Tanaka, J. What is "special" about face perception? *Psychological Review* 105. 3 (July 1998): 482-498
- [9] Geiger, D., Landendorf, B., and Yuille, A. (1995). Occlusions and binocular stereo. *International Journal of Computer Vision*, 14(3): 211-226.
- [10] Henderson, J. M., Brockmole, J. R., Castelhana, M. S., & Mack, M. (in press). Image salience versus cognitive control of eye movements in real-world scenes: Evidence from visual search. In R. van Gompel, M. Fischer, W. Murray, & R. Hill (Eds.), *Eye movement research: Insights into mind and brain*. Elsevier.
- [11] Ittli, S. and Bobick, A. (1994, pages = 179-186). Disparity-space images and large occlusion stereo. In *Proceedings of the 3rd European Conference on Computer Vision*.
- [12] Itti, L., Christof Koch, Ernst Niebur: A Model of Saliency-Based Visual Attention for Rapid Scene Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 20(11): 1254-1259 (1998)
- [13] Marques, O.; Mayron, L. M.; Socek, D.; Borba, G. B. & Gamba, H. R. An attention-based method for extracting salient regions of interest from stereo images International Conference on Computer Vision Theory and Applications (VISAPP), 2007
- [14] Navalpakkam, V. & Itti, L. Modeling the influence of task on attention *Vision Research*, 2005, 45, 205-231
- [15] Nelson, C.A. (2001) The development and neural bases of face recognition. *Infant and Child Development*, 10, 3-18.
- [16] Noton, D. & Stark, L. Scanpaths in Eye Movements during Pattern Perception *Science*, 1971, 171, 308-311
- [17] Oliva, A. Itti, L.; Rees, G. & Tsotsos, J. (ed.) *Gist of a Scene*, *Neurobiology of Attention*, Academic Press, Elsevier, 2005
- [18] Palmer, S. E. (1999a) *Vision science: Photons to phenomenology*. Cambridge, MA: Bradford Books/MIT Press.

- [19] Parkhurst, D.; Law, K. & Niebur, E. Modeling the role of salience in the allocation of overt visual attention *Vision Research*, 2002, 42, 107-123
- [20] Parkhurst, D. & Niebur, E. Texture Contrast Attracts Overt Visual Attention in Natural Scenes *European Journal of Neuroscience*, 2004, 19, 783-789
- [21] Peters, R.; Iyer, A.; Itti, L. & Koch, C. Components of bottom-up gaze allocation in natural images *Vision Research*, 2005, 45, 2397-2416
- [22] Pylyshyn, Z. W. *Seeing and Visualizing: It's Not What You Think* MIT Press, 2006
- [23] Rutishauser, U.; Walther, D.; Koch, C. & Perona, P. Is Bottom-Up Attention Useful for Object Recognition *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004, 11-37
- [24] Rybak, I. A., Gusakova, V. I., Golovan, A. V., Podladchikova, L. N., and Shevtsova, N. A. (1998) A model of attention-guided visual perception and recognition. *Vision Research* 38: 2387-2400.
- [25] Saygin, A.P., Cicekli, I. and Akman, V. (2000), 'Turing Test: 50 Years Later', *Minds and Machines* 10(4): 463-518.
- [26] Stentiford, F. An attention-based similarity measure with application to content-based information retrieval *Proc. of the Storage and Retrieval, Proc. of the Storage and Retrieval for Media Databases Conference, SPIE Electronic Imaging*, 2003
- [27] Styles, E. A. *Attention, Perception, and Memory: An Integrated Introduction* Taylor & Francis Routledge, 2005
- [28] Walther, D.; Itti, L.; Riesenhuber, M.; Poggio, T. & Koch, C. Attentional Selection for Object Recognition -- a Gentle Way *Lecture Notes in Computer Science*, 2002, 2525, 472-479
- [29] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics — Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, Aug. 2004.
- [30] Birchfield, S. and Tomasi, C. (1998). A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401-406.
- [31] Birchfield, S. and Tomasi, C. (1999). Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269-293.



[32] Intille, S. and Bobick, A. (1994). Disparity-space images and large occlusion stereo. In Proceedings of the 3rd European Conference on Computer Vision, pp. 179-186.

[33] Oge Marques, Liam M Mayron, Gustavo B Borba, and Humberto R Gamba, "An attention-driven model for grouping similar images with image retrieval applications", Eurasip Journal on Advances in Signal Processing, Special Issue on Image Perception, Vol. 2007, Article ID 43450, 17 pages, 2007.

## **4.8 Summary of Contributions and Deliverables**

In this section the second year research contributions and deliverables for stereo and multi-view image and video stabilization, calibration, coding, analysis and playback are identified.

### **4.8.1 Video Stabilization Contributions and Deliverables**

The objective of our study on Video Stabilization is to implement a software system for effectively reducing undesirable motion effects in coastline surveillance videos. Videos taken by hand or from mobile platforms often suffer from undesirable motion effects, which are caused by the unwanted motions of cameras. In addition, surveillance cameras mounted on static poles or platforms are also subject to atmospheric disturbances. As a result, the visual quality of collected videos is degraded. The objective of video stabilization, also known as image sequence stabilization (ISS), is to remove undesirable motion effects so that only intentional motion effects are retained.

The primary benefit of video stabilization is improving video quality, and in the context of surveillance applications, resulting in better performance measured by receiver operating characteristics (ROC). In addition, video stabilization has a desirable side effect of reducing the bit rate for encoding the stabilized videos.

Our contributions are listed as follows:

1) Previous studies use peak signal-to-noise ratio (PSNR) to evaluate the accuracy of motion estimation algorithms. This measure has several shortcomings. First, PSNR does not consider non-overlapping regions. Second, PSNR is only an indirect measure of the accuracy of estimated motion parameters.

We propose a measure “average pixel deviation” (APD) to directly assess the accuracy of estimated motion parameters in comparison to true motion parameters, which is capable of overcoming the aforementioned shortcomings.

2) A practical issue of error accumulation often arises during the computation of the transformation matrix between the current frame and the reference frame, which has not been addressed in the previous studies to the best of our knowledge. We propose a novel periodic correction strategy, which is capable of effectively reducing error accumulation.

3) In the context of developing a real-time video stabilization algorithm, we present a comparative study of four motion estimation methods, in the aspects of computational speed and accuracy.

4) In the context of developing a real-time video stabilization algorithm, we present a comparative study of the accuracy of three motion correction methods.

The deliverables of this project include the following items:

- Technical report that describing in detail our research methodology, results, and contributions
- Software programs that implement the proposed video stabilization algorithm
- Submitted a conference paper to IEEE IRI 2007 based on our research results

#### **4.8.2 Camera calibration and Stereo Correspondence Contributions and Deliverables**

Camera calibration is an important issue in computer vision since it is related to many vision problems such as stereovision. Camera calibration consists in the estimation of a model for an un-calibrated camera. The objective is to find the external parameters (i.e. position and orientation relatively to a world co-ordinate system), and the internal parameters of the camera (i.e., principal point or image centre, focal length and distortion coefficients). Good camera calibration is important when we need to reconstruct a world model or interact with the world, e.g., robot, hand-eye coordination etc. We survey on representative research papers on camera calibration, including Tsai's camera model and Zhang's flexible technique of camera calibration, and give a step-by-step usage of the practical camera calibration Toolbox from Caltech.

Local stereo correspondence is usually not satisfactory because neither big window nor small window based methods can accurately match densely-textured and textureless regions at the same time. We present a progressive edge-based stereo matching algorithm, in which big window and small window based matches are progressively integrated based on the edges of disparity map of a big window based matching. In addition, an arbitrarily-shaped window based matching is used for the regions where big windows and small windows can not find matches, and a novel optimization method, Progressive Outlier Remover, is used to remove outliers and noise. Empirical results show that our algorithm is comparable to some state-of-the-art stereo correspondence algorithms.

The following is the summary of contributions regarding camera calibration and stereo correspondence:

1) Surveyed research papers on camera calibration and used well-known camera calibration toolbox.

2) Proposed and implemented two stereo correspondence algorithms: (1) An arbitrarily-shaped window based stereo matching using a go-light optimization algorithm, and (2) a progressive edge-based stereo correspondence method.

The deliverables are as follows:

- Technical report that describes in detail our research methodology, results and contributions.
- Software programs that implemented the proposed stereo correspondence algorithms.
- Submitted two conference papers to IEEE ICIP 2007 and IEEE ICCV based on our research results.

The following is the summary of contributions regarding camera calibration and stereo correspondence:

#### **4.8.3 3D Video Coding and Playback Contributions and Deliverables**

It is anticipated that 3D video improves surveillance applications. An efficient compression of stereo view or multi-view sequences is a challenging task that we explored and for which we designed effective approaches. Inexpensive 3D Sharp autostereoscopic displays are used for testing.

Our contributions are listed as follows:

- 1) Player to play 3D video on autostereoscopic displays is developed. It is designed for Sharp 3D Displays. The player is developed using AVISynth and DirectShow libraries.
- 2) An algorithm for efficient compression of multi-view sequences using asymmetric video coding has been developed. The key point of the proposed algorithm is that one of the stereo views is coded at lower quality which does very little to affect the perception quality.
- 3) A hypercube predictive coding has been designed for efficient multi-view video coding.

The deliverables for this part of the project are:

- Technical report that describes in detail our research methodology, results and contributions.
- 3D video player for autostereoscopic displays capable of playing prerecorded stereo sequences (source code, executable, and documentation)
- 3D video player for autostereoscopic displays capable of playing live feed stereo camera pair (source code, executable, and documentation)

#### 4.8.4 Object Segmentation Using Depth Information Contributions and Deliverables

Attention-based ROI extraction based on two complementary computational models of human visual attention are proposed. Depth estimation using pixel-to-pixel stereo correspondence method is determined to suit well due to fast performance and good depth estimation. Salient objects (ROIs) are segmented with depth information, allowing for improved segmentation.

The following is a list of contributions:

- 1) Surveyed object segmentation and depth estimation research literature
- 2) Effective algorithms for object segmentation using depth information are developed and tested: (1) in combination with a biologically inspired methodology for ROI extraction, and (2) in combination with Bayesian-based foreground/background classifier for video object segmentation.
- 3) A number of experimental stereo sequences taken to test the proposed methodology.

Finally, the deliverables for this part of the project are as follows:

- Technical documentation describing in detail our research methodology, algorithms, results and contributions.
- Executables, sample sequences, source code and documentation for the proposed algorithms
- The new method (and associated experimental results) appeared in:
  - Oge Marques, Liam M. Mayron, Daniel Socek, Gustavo B. Borba and Humberto R. Gamba, “An attention-based method for extracting salient regions of interest from stereo images”, in *International Conference on Computer Vision Theory and Applications (VISAPP)*, March 8-11, 2007.
- Extended (journal) paper with additional results and extension to video being prepared.